Model 990 Computer
Data Dictionary User's Guide

Part No. 2276582-9701 *A
15 August 1982

TEXAS INSTRUMENTS

MANUAL REVISION HISTORY

Model 990 Computer Data Dictionary User's Guide (2276582-9701)

Original Issue ..... 15 October 1981
Revision ..... 15 August 1982

The total number of pages in this publication is 156.

Preface

This manual contains information about the Texas Instruments Data Dictionary (DD-990), which operates under the Texas Instruments Model 990 Computer. Both inexperienced users and experienced programmers can use this information. The user should have access to a fully operational system with the System Command Interpreter (SCI), a Model 911 Video Display Terminal (VDT) in video display mode, and a valid ID and passcode if security has been installed on the system. This manual is organized as follows:

Section

1 General Description — Provides an overview of DD-990 and explains the concepts, functions, environments, and features of the system.

2 DD-990 Configuration — Discusses the options and considerations in the DD-990 configuration.

3 DD-990 File Maintenance — Describes the procedures and options necessary to create, assign, release, restore, save, and print the contents of a data dictionary file.

4 DD-990 Data Librarian — Discusses the Interactive Data Librarian (IDL) and the Batch Data Librarian (BDL).

5 DD-990 Utilities — Discusses reporting functions and automatic file definition capabilities.

6 Data Manager and Query-990 — Discusses the benefits of using Query-990 in conjunction with DD-990.

7 Security — Discusses the security options and features for Query-990 and DD-990.

8 Error Messages — Explains the error messages associated with DD-990.

Appendix

A DD-990 DDL Syntax and Key Words — Provides the data file syntax and key word definitions of data files for DD-990.

B DD-990 Function Keys and Data Types — Explains the function keys and data types for DD-990.

C DD-990 Test File Listings — Lists the descriptions and data in files used in examples.

The following documents contain additional information related to operating DD-990:

Title	Part Number
Model 990 Computer DX10 Data Base Management System Programmer's Guide	2250425-9701
Model 990 Computer DX10 Data Base Administrator User's Guide	2250426-9701
Model 990 Computer DX10 Query-990 User's Guide	2250466-9701
Model 990 Computer DX10 Operating System Production Operation Manual, Volume II	946250-9702
Model 990 Computer DX10 Operating System Error Reporting and Recovery Manual, Volume VI	946250-9706
Model 990 Computer DNOS Data Base Administrator User's Guide	2272059-9701
Model 990 Computer DNOS Data Base Management System Programmer's Guide	2272058-9701
Model 990 Computer DNOS Query-990 User's Guide	2276554-9701
Model 990 Computer DNOS Operations Guide	2270502-9701
Model 990 Computer DNOS Messages and Codes Reference Manual	2270506-9701

Contents

5 — DD-990 Utilities

Appendixes

Index

Illustrations

Tables

1

General Description

1.1 INTRODUCTION

As information systems expand, the need for accuracy and control of data becomes increasingly important. Data as an asset and a resource requires some form of standards enforcement. The Texas Instruments Data Dictionary (DD-990) system allows you to define all the data used in an organization and store these definitions in a central location. This centralization aids in the enforcement of data standards, clarifies the impact of changes to the data, and limits data redundancy.

This section provides an overview of the DD-990 system, describes the environments and capabilities, and defines the terminology used in this manual. Readers should have an operational knowledge of the Texas Instruments Model 990 Computer and the operating system. A working knowledge of Query-990 and DBMS-990 is also required if either system is installed.

1.2 DD-990 OVERVIEW

The DD-990 system consists of a dictionary file, a data librarian, utilities, and a data manager. DD-990 maintains and coordinates diverse kinds of files and programs through the dictionary file.

The dictionary file contains the definitions of data in other files. (Note that the dictionary file does not include the actual data contained in files.) Four kinds of files are controlled and maintained:

- Key indexed
- Relative record
- Sequential
- Data base

The dictionary file contains definitions of data files and describes the kinds of data entities in the file. The dictionary file consists of two major parts, entities and attributes. Entities are fields, groups, files, and programs. They are identified in the dictionary file by four-character unique IDs. Attributes consist of all other elements of the dictionary file: alternate names, secondary keys, tags, and pathnames.

You can enter information into the dictionary file through the data librarian. The data librarian checks all the file definitions in the dictionary file. The data librarian has the ability to accept information in both interactive and batch mode. The data librarian performs extensive checking before any definitions are entered in the dictionary file.

The DD-990 utilities generate detail, summary, and cross-reference reports. This is an effective means of keeping up with file definitions in the dictionary file and understanding the relationships of the definitions to each other.

The data manager provides the DD-990 interface to Query-990. The data manager allows Query-990 to access the conventional file types. The possibilities for inquiry and report generating are unlimited in this environment.

1.3 ENVIRONMENTS AND CAPABILITIES

DD-990 operates on the DS990 Model 4 (or larger) Computer, under the DX10 or DNOS operating system. The minimum amount of memory required for DD-990 under DX10 is 192K bytes. The minimum amount of memory required for DD-990 under DNOS is 256K bytes. DD-990 provides many capabilities in minicomputer environments which can be used by both data processing and non-data-processing personnel with minimal training.

DD-990 can be configured and tailored to the requirements of the following environments:

- As a stand-alone system
- With Query-990
- With the Data Base Management System (DBMS)

The capabilities of DD-990 vary with the particular environment in which the system is installed. Some of the features of DD-990 as a stand-alone system are as follows:

- Generates summary, detail, and cross-reference reports about the data entities and attributes defined in the dictionary file
- Generates COBOL record descriptions for use in the FILE SECTION of COBOL programs
- Generates data definition language DDL — descriptions for any file previously defined in the dictionary file
- Allows the List File Utility to generate a DDL description for a file previously defined in the dictionary file.
- Allows security to be established on conventional and data base files.
- Accepts DDL descriptions through the Batch Data Librarian (BDL)
- Creates and maintains dictionary definitions using the Interactive Data Librarian (IDL)

When used in conjunction with Query-990, DD-990 allows nonprocedural access to conventional files. This feature permits users to perform simple to complex queries on conventional files without extensive programming. Queries can be made interactively or can be embedded in an application program.

When used with DBMS-990, DD-990 provides the definitions for data base files.

DD-990 also provides the Data Administrator with accurate and complete information about the structure of data. This information allows the definition and enforcement of data standards.

1.4 TERMINOLOGY

When defining entities and attributes to the dictionary file, it is important to have an understanding of the DD-990 system terminology. DD-990 describes data in the following terms:

- File — A collection of records. DD-990 supports four types of files: key indexed, relative record, sequential, and data base.
- Record — A collection of all lines whose primary keys have the same value.
- Line — A collection of fields and/or groups of fields. A line should define the format of the data normally retrieved by a single I/O operation in an application program. Lines are identified by a two-character ID, which must be unique within a single file definition.
- Key — A field whose value can be used to directly access lines having the same value of that field. In relative record and sequential files, the primary key is always the record number.
- Group — A collection of contiguous fields within a line.
- Field — A named data element in a line.
- Tag — A field whose value determines which line type to use to format a particular line of data in a file. A tag field is a fixed length string that occurs at the same offset in every line in a file. Its value is the same for all lines of a particular type in a file.
- ID — A four-character name that uniquely defines an entity in the dictionary file.
- Alternate name — A unique name of up to 30 characters in length by which an entity may be referenced instead of the ID.

2

DD-990 Configuration

2.1 INTRODUCTION

The configuration of DD-990 involves generating and installing the entire DD-990 system. The configuration is a straightforward process involving installed software, file types, and security. Figure 2-1 gives an overview of the entire configuration process including the security and Query-990 options.

Throughout this manual, screens are followed by a discussion of the possible responses to the prompts on that screen. The terms *target disk* and *target system* refer to the disk on which the DD-990 system will be installed. Before proceeding with the DD-990 generation, create a backup copy of the installation disk.

2.2 DD-990 GENERATION

To initiate the DD-990 generation process, enter the DDGEN command. The first configuration screen appears:

```
DD-990 GENERATION <VERSION L.V.E YYDDD>

                    DD-990 PATHNAME:  DD990
DBMS-990 ON TARGET SYSTEM(Y/N):  NO
  TIFORM ON TARGET SYSTEM(Y/N):  NO
```

Respond to the prompts as follows:

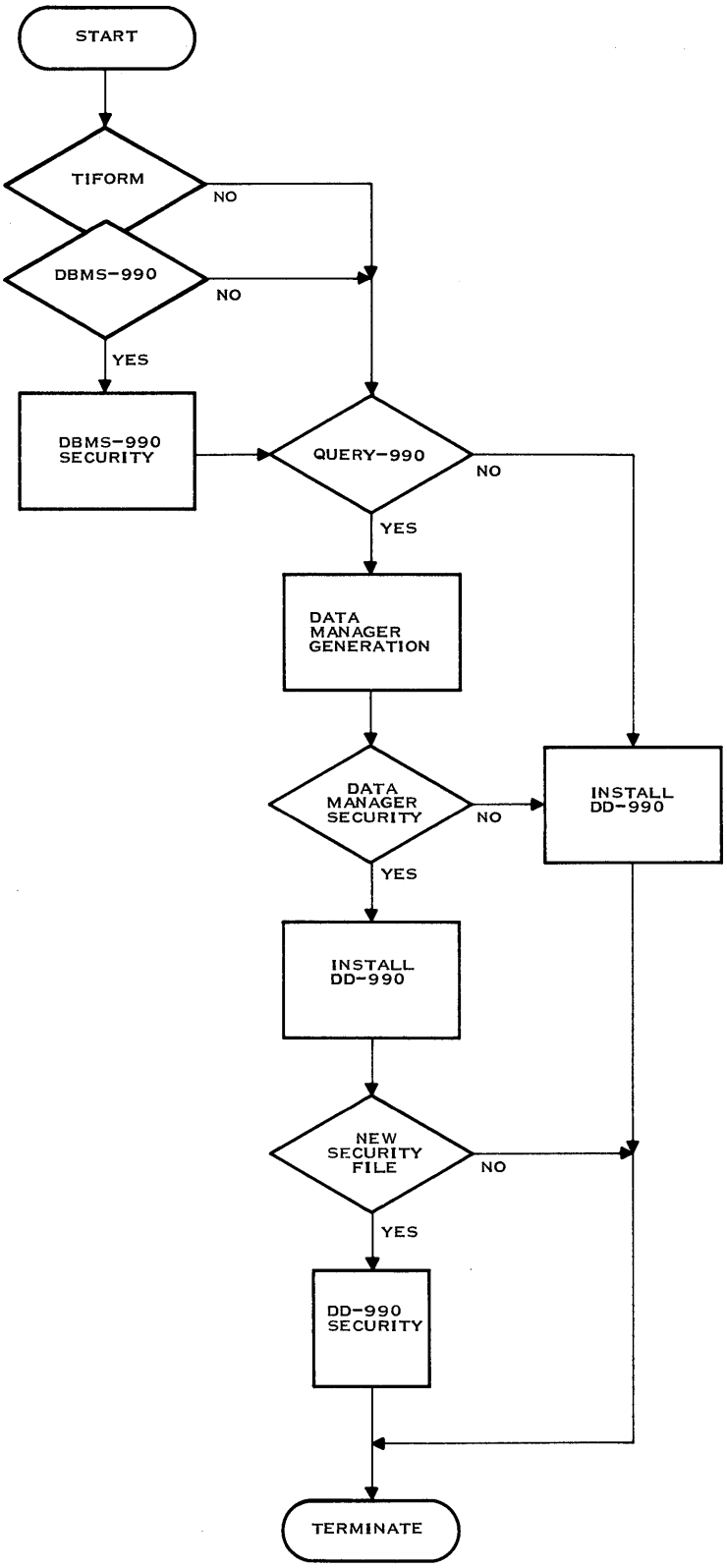
DD-990 PATHNAME
: Enter the name of the DD-990 installation disk or the DD-990 directory pathname. This process writes to the installation disk; therefore, the installation disk must not be write protected.

DBMS-990 ON TARGET SYSTEM(Y/N)
: Enter Y to this prompt if DBMS-990 is installed on the target system (see paragraph 2.3). Otherwise, enter N. The default is N.

NOTE

If DBMS has been previously installed, DD-990 generation process will delete the installed version of DBMS if N is answered for the DBMS-990 ON TARGET SYSTEM prompt.

TIFORM ON TARGET SYSTEM(Y/N)
: Enter Y if TIFORM is installed on the target system. Otherwise, enter N. The default is N.



2280556

Figure 2-1. DD-990 Generation Process

2.3 DBMS-990 CONFIGURATION

If DBMS-990 is installed on the target system, and the prompt DBMS-990 ON TARGET SYSTEM(Y/N) was answered Y, the following screen appears:

```
DBMS-990 CONFIGURATION

  INCLUDE DBMS-990 SECURITY?(Y/N):  YES
```

Respond to the prompt as follows:

INCLUDE DBMS-990 SECURITY?(Y/N)
Enter Y if DBMS-990 was generated on the target disk with security. Otherwise, enter N. The default is Y. DBMS-990 security is to some extent independent of the DD-990 security discussed later in this section. It is possible to have security on DBMS-990 without having security on DD-990. In this case, unauthorized access is only prevented on database files. If security is included on DD-990, unauthorized access is prevented on conventional files when accessed through Query-990 (see Section 7).

The choice of the security option requires special considerations. If DBMS-990 exists with security on the target disk and the security option for DD-990 also is selected, the DBMS security file will exist for DD-990.

2.4 QUERY-990 OPTION

If Query-990 has been installed on the system, the following screen will appear:

```
QUERY-990

        QUERY-990 ON TARGET
              SYSTEM?(Y/N):  YES
```

Respond to the prompt as follows:

QUERY-990 ON TARGET SYSTEM?(Y/N)
Enter Y if QUERY-990 is or will be installed on the target system. Otherwise, enter N. The default is Y.

The screen for the data manager option appears only if you entered Y in response to the prompt QUERY-990 ON TARGET SYSTEM?(Y/N). When answering the prompts regarding file types, do not select file types that are not used on the target system. Selecting file types not used on the target system increases the size of the data manager and requires more memory during execution.

The screen for data manager generation appears as follows:

```
DATA MANAGER GENERATION

          KEY INDEXED FILE ACCESS:  YES
    RELATIVE RECORD FILE ACCESS:  YES
          SEQUENTIAL FILE ACCESS:  YES
    DATA MANAGER SECURITY?(Y/N):  NO
```

Respond to the prompts as follows:

KEY INDEXED FILE ACCESS
Enter Y if key indexed files (KIFs) are to be queried. Otherwise, enter N. The default is Y.

RELATIVE RECORD FILE ACCESS
Enter Y if relative record files are to be queried. Otherwise, enter N. The default is Y.

SEQUENTIAL FILE ACCESS
Enter Y if sequential files are to be queried. Otherwise, enter N. The default is Y.

DATA MANAGER SECURITY?(Y/N)
Enter YES if the data manager is to be generated with security. Otherwise, enter N. If you select Y, passwords will be required in order to query conventional files.

When you have answered all prompts, DDGEN continues as a background task for about 20 minutes. When the DD-990 generation completes successfully, the following message appears:

```
DDGEN complete with 0 errors
```

If errors occur during DD-990 generation, instructions for error recovery are printed on the error message listing.

2.5 DD-990 INSTALLATION

Once the DD-990 system has been generated, it is ready to be installed on the system disk. Enter the command DDINS to install DD-990. The following screen appears:

```
INSTALL DD-990

                DD-990 PATHNAME:  DD990
                    TARGET DISK:
NEW SECURITY FILE?(Y/N):  YES
```

Respond to the prompts as follows:

DD-990 PATHNAME
Enter the name of the DD-990 installation disk or the DD-990 directory pathname.

TARGET DISK
Enter the volume name of the system disk. This can be a disk other than the current system disk. This DD-990 installation writes on the target disk; therefore, the target disk cannot be write protected.

NEW SECURITY FILE?(Y/N)
This prompt appears only if DD-990 was generated with the security option. Enter Y if you want to replace all existing security files; in this case, any existing DBMS-990 security files are deleted (see paragraph 2.6). The security prompt will not appear if security has not been installed.

2.6 DD-990 SECURITY

If you entered Y in response to the NEW SECURITY FILE?(Y/N) prompt, the following screen appears:

```
DD-990 SECURITY

          MAXIMUM ENTRIES:
       MAXIMUM PASSWORDS:
         MASTER PASSWORD:
```

Respond to the prompts as follows:

MAXIMUM ENTRIES
: Specify the maximum number of fields, groups, lines, and files to which passwords are to be assigned.

MAXIMUM PASSWORDS
: Specify the maximum number of passwords to be assigned.

MASTER PASSWORD
: Enter the four-character alphanumeric value to be used as the master password for the installed system.

Refer to Section 7 for a thorough discussion of DD-990 security.

3

DD-990 File Maintenance

3.1 INTRODUCTION

DD-990 includes a number of commands that allow you to maintain the dictionary file. This section describes these commands and explains their interrelationships and how to use them. The file maintenance commands are as follows:

- Create Data Dictionary (CDD) — Creates a new dictionary file tailored to specified sizes
- Assign Data Dictionary (ADD) — Assigns the dictionary file so that the utilities can be performed
- Release Data Dictionary (RDD) — Releases the dictionary file
- Data Dictionary Status (DDSTAT) — Generates listing of estimated and actual values assigned to the dictionary file during configuration
- Save Data Dictionary (DDSAVE) — Creates a backup of the existing dictionary file
- Restore Data Dictionary (DDRSTR) — Replaces the dictionary file on the system

The purpose of the file maintenance commands is to manipulate the dictionary file. Once DD-990 has been installed, the CDD command is used to create the dictionary file. Regardless of the functions to be performed on the dictionary file, the file must first be assigned (ADD command). The DDSAVE command should be used to maintain a current backup of the dictionary file. In situations where more than one dictionary file is required, the RDD command can be used to release the current dictionary file before assigning another. For example, if too much or too little space is allocated to the dictionary file, the RDD command releases the current file so that you can modify previous space allocations and create a new dictionary file.

NOTE

If DBMS has been previously installed on the system that will receive DD-990, the data base manager must be running. Use the Start Data Base (SDBMS) command to start the data base manager.

3.2 CREATE DATA DICTIONARY — CDD

The Create Data Dictionary (CDD) command creates a new dictionary file. This command requires specific knowledge of the data environment. To determine the amount of space needed in the dictionary file, first count the number of fields, groups, files, and programs that presently exist. Estimate the additional number of fields, groups files, and programs that will be needed. Then combine the existing and estimated numbers of existing entities. Enter the CDD procedure to create the data dictionary file. The following screen appears:

```
CREATE DATA DICTIONARY <VERSION L.V.E. YYDDD>

                                MASTER PASSWORD:
                                      MAX FIELDS:
                                     MAX GROUPS:
                                      MAX FILES:
                                   MAX PROGRAMS:
                                      MAX NAMES:
                                MAX DESCRIPTIONS:
                                AVG FIELDS/GROUP:
                          AVG FIELDS + GROUPS/FILE:
                                AVG LINE TYPES/FILE:
                               DICTIONARY PATHNAME:
                          INITIALIZE NEW DICTIONARY?(Y/N):
```

Respond to the prompts as follows:

MASTER PASSWORD
: Enter the DBMS-990 master password. The password prompt appears only if DBMS-990 with security exists on the target system.

MAX FIELDS
: Enter the maximum number of fields to be defined in the dictionary file.

MAX GROUPS
: Enter the maximum number of groups to be defined in the dictionary file.

MAX FILES
: Enter the maximum number of files to be defined in the dictionary file.

MAX PROGRAMS
: Enter the maximum number of programs to be defined in the dictionary file.

MAX NAMES
: Enter the maximum number of alternate and primary names to be used in the data environment. This includes file, field, group, line, and program names.

MAX DESCRIPTIONS
: Enter the maximum number of lines of description to be used in the data environment.

AVG FIELDS/GROUPS
Enter the average number of fields per group to be used in the data environment.

AVG FIELDS + GROUPS/FILE
Enter the average number of fields and groups per file to be used in the data environment.

AVG LINE TYPES/FILE
Enter the average number of line types per file to be used in the data environment.

DICTIONARY PATHNAME
Enter the pathname where the dictionary file will reside. The recommended pathname is .S$DD.DDF$. This file cannot already exist on the target system.

INITIALIZE NEW DICTIONARY?(Y/N)
Enter Y when the dictionary file is being created for the first time. A Y response prohibits the DDRSTR command from restoring previously saved contents into the new dictionary file.

Enter N when the dictionary file is being expanded or compressed. Enter N only when you plan to restore the dictionary file (DDRSTR).

3.3 ASSIGN DATA DICTIONARY — ADD

The ADD command associates the dictionary file with a pathname. You cannot access the dictionary file until it has been assigned. The ADD command allows the use of other DD-990 utilities. Until the dictionary file has been assigned, no operations can be performed on the file. Enter the ADD command to assign the dictionary file. The following screen appears:

```
ASSIGN DATA DICTIONARY <VERSON L.V.E YYDDD>

    DICTIONARY PATHNAME:
```

Respond to the prompt as follows:

DICTIONARY PATHNAME
Enter the pathname where the dictionary file is stored. The recommended pathname is .S$DD.DDF$.

3.4 RELEASE DATA DICTIONARY — RDD

The RDD command releases the dictionary file. Several dictionary files can exist on the system. However, only one dictionary file can be assigned and used at one time. Also, one dictionary file cannot be assigned while another dictionary file is being created. This command releases the LUNO associated with the currently assigned dictionary file. The following screen appears:

```
RELEASE DATA DICTIONARY <VERSION L.V.E YYDDD>

        ARE YOU SURE(Y/N)?:
```

Respond to the prompt as follows:

ARE YOU SURE(Y/N)?
Enter Y if the dictionary file is to be released. Otherwise, enter N.

NOTE

Be sure that the dictionary file is to be released before entering Y in response to this prompt. Once the dictionary file has been released, no activity can be performed until the dictionary file has been reassigned.

3.5 DATA DICTIONARY STATUS — DDSTAT

The Data Dictionary Status (DDSTAT) command is a measurement facility for the dictionary file. The DDSTAT command compares the actual size of the dictionary file with the maximum size specified in the CDD command by listing both the specified storage requirements and the actual amount of storage currently used. To generate the dictionary file status report, enter the DDSTAT command. The following screen appears:

```
DATA DICTIONARY STATUS <VERSION L.V.E YYDDD>

          MASTER PASSWORD:
    LISTING ACCESS NAME:
               MODE (F, B):  B
```

Respond to the prompts as follows:

PASSWORD
This prompt appears only if DBMS-990 with a security file exists on the target system. Enter a previously assigned master password.

LISTING ACCESS NAME
Enter the name of a device where the listing of statistics can be output.

MODE (F, B)
Enter B to execute DDSTAT in background mode. Enter F to execute DDSTAT in foreground mode.

The DDSTAT command produces a report similar to that shown in Figure 3-1. The report compares the estimations entered using the CDD command with the actual number of entities in the dictionary file. The DDSTAT command records the original specifications under the MAX ENTRIES column and records the actual specifications under the TOTAL ENTRIES column. Notice the AVERAGE FIELDS and AVERAGE LINE statistics. DD-990 calculates these averages from the actual dictionary file and displays them under the TOTAL column.

This report is useful when reallocating the dictionary file. It is also an excellent method of tailoring the dictionary file to avoid wasting disk space.

DD-990 STATUS REPORT L.V.E YY.DDD MM/DD/YY HH:MM:SS

DATA DICTIONARY CHARACTERISTICS	ENTRIES MAX	TOTAL
FIELDS:	300	235
GROUPS:	20	13
FILES:	200	35
PROGRAMS:	5	0
NAMES:	25	9
DESCRIPTIONS:	25	8
AVERAGE FIELDS/GROUP:	4	1
AVERAGE FIELDS + GROUPS/FILE:	15	9
AVERAGE LINE TYPES/FILE:	4	1

Figure 3-1. DDSTAT Report

3.6 SAVE DATA DICTIONARY FILE — DDSAVE

The DDSAVE command copies the dictionary file. You should make a backup copy of the dictionary file on a regular basis. The dictionary file to be backed up must be assigned. To backup the assigned dictionary file, enter the DDSAVE command. The following screen appears:

```
SAVE DATA DICTIONARY <VERSION L.V.E YYDDD>

          MASTER PASSWORD:
DICTIONARY PATHNAME:
  SAVE FILE PATHNAME:
```

Respond to the prompts as follows:

MASTER PASSWORD
: This prompt appears only if you selected the DBMS-990 security option. Enter the DBMS-990 master password.

DICTIONARY PATHNAME
Enter the pathname where the dictionary file is stored. The dictionary file must be assigned.

SAVE FILE PATHNAME
Enter the pathname where the backup copy of the dictionary file will reside.

3.7 RESTORE DATA DICTIONARY — DDRSTR

The DDRSTR command restores the saved contents of a dictionary file to a new dictionary file. To restore the dictionary file, enter the DDRSTR command. The following screen appears:

```
RESTORE DATA DICTIONARY <VERSION L.V.E YYDDD>

          MASTER PASSWORD:
DICTIONARY PATHNAME:
  SAVE FILE PATHNAME:
```

Respond to the prompts as follows:

MASTER PASSWORD
This prompt appears only if DBMS-990 with a security file exists on the target system. Enter the assigned master password.

DICTIONARY PATHNAME
Enter the pathname where the dictionary file is stored. The dictionary file must be assigned.

SAVE FILE PATHNAME
Enter the pathname where the backup copy of the dictionary file has been stored.

The actual amount of storage required for the dictionary file can be difficult to estimate accurately. In the case where the dictionary file is to be reallocated, the sequence of the file maintenance commands is extremely important. Figure 3-2 shows the steps involved in the process of reallocating a dictionary file. Notice the order of the commands as the dictionary file is modified.

3.8 REALLOCATING A DICTIONARY FILE

Figure 3-2 illustrates the steps necessary to reallocate a dictionary file. Determine the amount of space currently used in the dictionary file using the DDSTAT command. Back up the existing dictionary file using the DDSAVE command. Release and delete the current dictionary file using the RDD command. Use the Modify File Name (MFN) command if you wish to rename instead of deleting the current dictionary file. Create and assign the new dictionary file using the CDD and ADD commands. Restore the contents of the old dictionary file, and back up the new dictionary file.

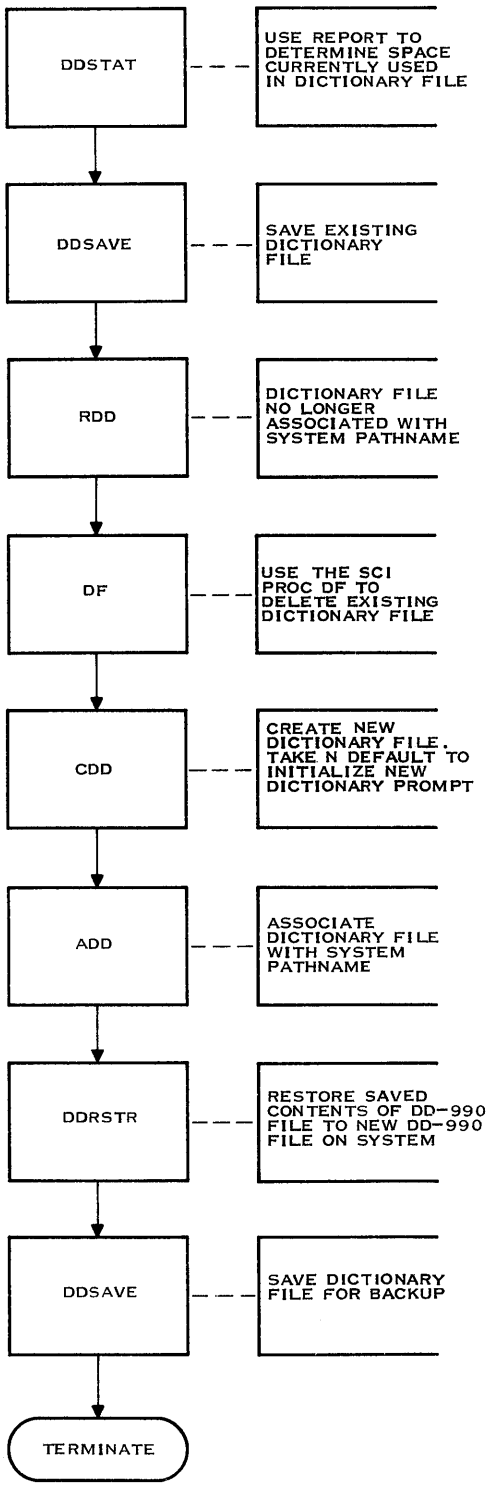


Figure 3-2. Reallocating a Data Dictionary File

4

Data Librarian

4.1 INTRODUCTION

The data librarian is the DD-990 component that is responsible for the accuracy of all definitions in the dictionary file. You can use the data librarian to enter, update, and delete definitions in the dictionary file. All information considered for the dictionary file must meet the standards of the data librarian. These standards involve checking for redundancy, correct syntax, and status of the dictionary file.

The data librarian provides two types of access to the dictionary file: access through the Interactive Data Librarian (IDL) and access through the Batch Data Librarian (BDL).

To enter definitions into the dictionary file interactively, use the IDL procedure. This procedure is a menu-driven facility that reads information as it is entered on the screen. The IDL procedure verifies all definitions before they are sent to the dictionary file.

To enter definitions into the dictionary file using the BDL procedure, an input file is required. This input file must meet the format and key word specifications discussed in this section and summarized in Appendix A.

Regardless of the method of entry, DD-990 insures accurate and unique definitions in the dictionary file through the data librarian.

4.2 INTERACTIVE DATA LIBRARIAN — IDL

The interactive data librarian is a menu-driven facility that allows you to add, update, and delete definitions in the dictionary file. IDL is designed so that you can enter data through screens formatted to handle the necessary syntax required by DD-990.

The following paragraphs consist of a general discussion of the data hierarchy, IDL operations, and the general use of the function keys. Each IDL screen and the associated prompts will then be discussed. The screen operations and functions follow each appropriate screen.

4.2.1 Data Hierarchy

When you enter definitions into the dictionary file using IDL, a certain order is required for fields, groups, and files. The steps are as follows:

1. Define the fields in the dictionary file.

2. Define the groups. Groups contain previously defined fields.

3. Define the files. Files contain previously defined fields and/or groups.

4. Define programs. Programs can only reference previously defined files.

Alternate names can be defined at any time once the associated entity has been defined. Paragraph 4.2.10 discusses alternate names. Pathnames can also be defined at anytime once their associated files have been defined in the dictionary file.

Figure 4-1 illustrates the data hierarchy in a small dictionary file.

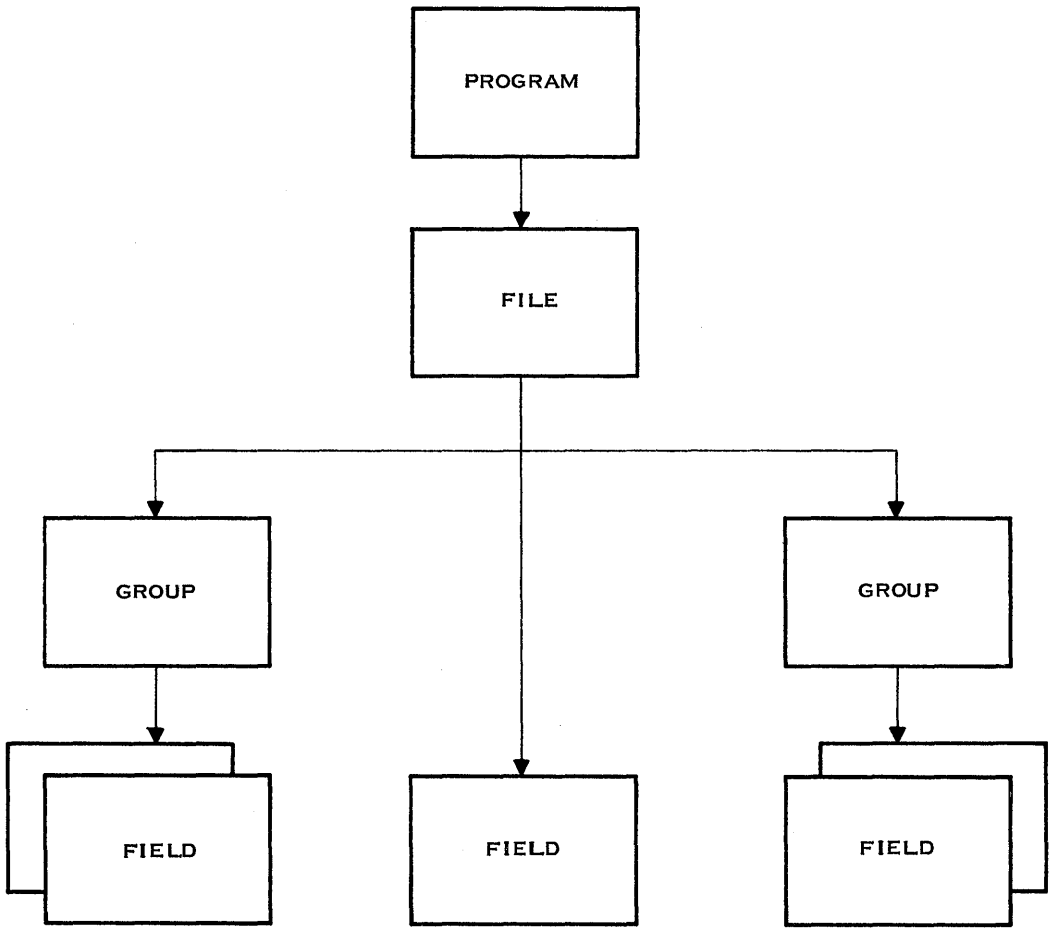


Figure 4-1. IDL Data Hierarchy

4.2.2 IDL Operations

The IDL performs the following operations:

- Add — When you enter a name or ID that does not currently exist in the dictionary file, the data librarian assumes the add mode.
- Update — When you enter a name or ID that already exists in the dictionary file, the data librarian assumes the update mode. The data librarian retrieves a current definition and displays the information on the screen.
- Delete — You can use the F7 key to delete data from the dictionary file whenever the data librarian is in update mode. There are instances where the position of the cursor is important for appropriate deletions.

4.2.3 Function Keys

The function keys have specific functions tailored to each IDL screen. The active function keys for each screen appear in the upper right-hand corner of the screen. Generally, the function keys perform the following operations in the interactive mode:

Key	Operation
F1	Roll up
F2	Roll down
F3	Inactive
F4	Inactive
F5	Scroll up
F6	Scroll down
F7	Delete
F8	Insert
CMD	Abort or return to main screen
ENTER	Enter data in dictionary file
PRINT	Print current screen
BACKFIELD	View data currently in dictionary file

The CMD key has two unique properties:

- When information is contained on the screen, the CMD key assumes the abort function.
- When no information is contained on the screen, the CMD key returns to the main IDL screen.

Appendix B lists and explains the function keys in detail.

4.2.4 IDL Procedure

To activate the data librarian in interactive mode, enter the IDL procedure. The following screen appears:

```
DATA LIBRARIAN <VERSION L.V.E YYDDD>

                    PASSWORD:
```

Respond to the prompt as follows:

PASSWORD
: This prompt appears only if DBMS-990 security has been installed. Enter the assigned master password.

The following screen appears for the IDL options. Select one of the options.

```
                        DD-990
                Interactive Data Librarian
                     Version L.V.E


Select one of the following:

        ALTERNATE NAME. . . . . . . . . . . (AN)
        FIELD. . . . . . . . . . . . . . . . (FL)
        FILE. . . . . . . . . . . . . . . . . (FI)
        GROUP. . . . . . . . . . . . . . . . (GR)
        PATHNAME. . . . . . . . . . . . . . . (PA)
        PROGRAM. . . . . . . . . . . . . . . (PR)
        QUIT. . . . . . . . . . . . . . . . . (QU)


Function keys:

   F1 - Roll up          F5 - Scroll up        F7 - Delete
   F2 - Roll down        F6 - Scroll down      F8 - Insert
   ENTER - Update DD     CMD - Abort           PRINT - Print screen
```

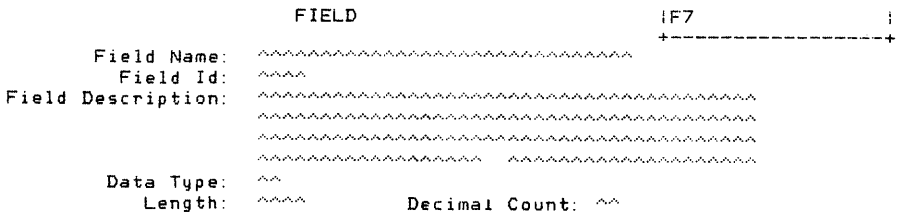
NOTE

The logical name S$TIFORM must be previously assigned under a DNOS system.

Once you have selected one of the options listed on the IDL screen, the appropriate screen appears. If the name of the ID you entered is already defined in the dictionary file, all the information stored for that particular entity will appear on the screen. At this time, any of the values associated with the entity can be changed. Press the ENTER key to include these values. The following paragraphs discuss the subsequent screens used to enter, update, or delete definitions in the dictionary file. The carats (^) on each screen represent the available length for responses to each prompt.

4.2.5 Field Screen

Once the field option has been selected from the original screen, the following screen appears for fields:



Respond to the prompts as follows:

Field Name

Enter a field name from 1 to 30 characters long to describe this field. Names must be unique throughout the data dictionary.

Field Id

Enter a name from 1 to 4 characters long to uniquely identify the field to the dictionary file. The first character must be alphabetic. The remaining characters can be alphabetic or numeric. The dictionary file can automatically assign the field ID. If you leave this field blank, the data librarian generates a unique ID beginning with A001.

Field Description

The field description is an optional documentation feature. You can store any textual description of the data in this field. This is an excellent means of internal documentation. The field description is included in detailed report listings.

Data Type

Enter a two-character code representing the overall characteristics of the data. Data types indicate whether data is character, numeric, or signed or has decimal positions. The data types are listed on the screen. See Appendix B for a detailed listing of the data types supported by DD-990.

Length

Enter the total number of characters allowed in the field. The length is automatically entered for the data types IS, ID, RS, RD, and LG.

Decimal Count

Enter the number of digits to be displayed to the right of the decimal point. The decimal count option occurs only for numeric noninteger data types. These data types include AS, AN, CS, CN, and PK. The default is zero.

4.2.5.1 Field Operations. Depending on whether the information already exists in the dictionary file, one of the following prompts appears when you press the ENTER key, or when you have responded to all previous prompts:

```
ADD THIS FIELD?(Y/N):

UPDATE THIS FIELD?(Y/N):
```

Respond to the prompts as follows:

ADD THIS FIELD?(Y/N)
If the information is new to the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

UPDATE THIS FIELD?(Y/N)
If the information is already contained in the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

The F7 key allows you to delete a field that is not referenced in a group or file in the dictionary file. When the F7 key is pressed, the following prompt appears:

```
DELETE THIS FIELD?(Y/N):
```

Respond to the prompt as follows:

DELETE THIS FIELD?(Y/N)
Enter Y if the field is to be deleted.

4.2.5.2 Field Function Keys. The following paragraphs describe the field function keys for the field screen:

F7 Key. This key allows you to delete a field that is not referenced elsewhere in the dictionary file. The following message appears when the field cannot be deleted:

```
ENTITY IS REFERENCED BY OTHER DICTIONARY
ENTITIES AND CANNOT BE DELETED.
```

CMD Key. When information is on the screen and the CMD key is pressed, the following prompt appears:

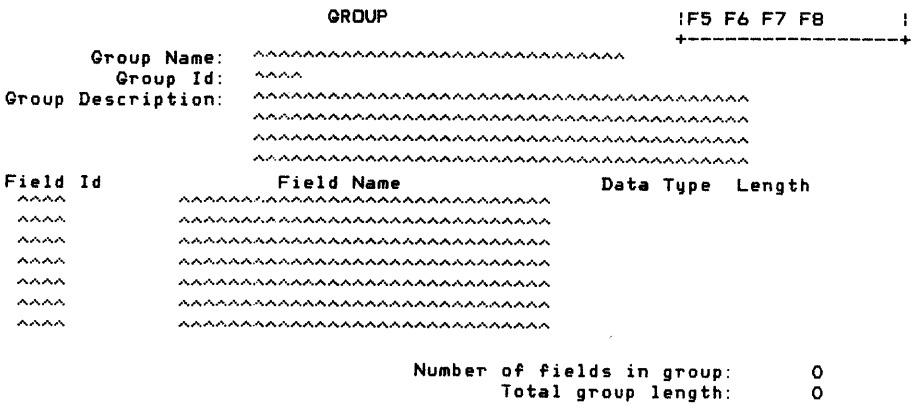
```
ABORT THIS SCREEN?(Y/N):
```

Enter Y in response to this prompt if the screen is to be aborted. Information is not sent to the dictionary file. Enter N to return to the original screen.

After you complete the field definition, the IDL includes the information in the dictionary file. The screen is then ready to accept another field definition. Use the CMD key to return to the original screen.

4.2.6 Group Screen

Once the group option has been selected from the original screen, the following screen appears for groups:



Respond to the prompts as follows:

Group Name

Enter a name from 1 to 30 characters long to describe one or more contiguous fields. Names must be unique throughout the dictionary file.

Group Id

Enter a name from 1 to 4 characters long to uniquely identify the group to the dictionary file. The first character must be alphabetic. The remaining characters can be alphabetic or numeric. If you leave this field blank, the data librarian generates a unique ID.

Group Description

The group description is an optional documentation feature. You can store any textual description of the data in this group. This is an excellent means of internal documentation. The group description is included in detailed report listings.

Enter either the field IDs or names that comprise this group. The field data type and length are displayed with each field. As you enter the field information, the data librarian computes the number of fields in the group and the total group length.

NOTE

A group cannot be defined within a group.

You can enter a total of seven fields on one screen. When you enter the seventh field, the screen continues to scroll up one line at a time to allow additional field entries for the group.

4.2.6.1 Group Operations. Depending on whether the information already exists in the dictionary file, one of the following prompts appears when you press the ENTER key:

ADD THIS GROUP?(Y/N):

UPDATE THIS GROUP?(Y/N):

Respond to the prompts as follows:

ADD THIS GROUP?(Y/N)
If the information is new to the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

UPDATE THIS GROUP?(Y/N)
If the information is already contained in the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

When the group is referenced elsewhere in the dictionary file, the data librarian does not allow access to other parts of the screen. In this case, the following message appears:

ONLY THE NAME AND DESCRIPTION CAN BE MODIFIED

The F7 key allows you to delete a group that is not referenced in a file in the dictionary file. When you press the F7 key, the following prompt appears:

DELETE THIS GROUP?(Y/N):

Respond to the prompt as follows:

DELETE THIS GROUP?(Y/N)
Enter Y if the group is to be deleted.

4.2.6.2 Group Function Keys. The following paragraphs describe the group function keys.

F5 Key. This key activates the scroll up function for the field information on the screen.

F6 Key. This key activates the scroll down function for the field information on the screen.

F7 Key. This key activates the delete function for groups that are not referenced in the dictionary file. The following message appears when the group cannot be deleted:

ENTITY MAY NOT BE DELETED

When the cursor is at the group name or ID, pressing the F7 key deletes the entire group from the dictionary file. When the cursor is at a field name or ID contained in the group, pressing the F7 key deletes the field within the group.

F8 Key. This key allows you to insert a field at a particular location in the group definition. If you press the F8 key and then decide the insert is not desired, press the F7 key to delete the insert.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. Information is not sent to the dictionary file. Enter N to return to the current screen.

After you complete the group definition, the IDL will include the information in the dictionary file. The screen is then ready to accept another group definition. Use the CMD key to return to the original screen.

4.2.7 File Screen

The file screen includes subsequent screens for lines and keys. Figure 4-2 illustrates the key options for the four kinds of files. Once you have selected the file option from the original screen, the following screen appears for files:

```
                                    FILE                  ¦F1 F7             ¦
                                                          +------------------+
                   File Name:  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                     File Id:  ^^^^
            File Description:  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                   File Type:  ^^

            Primary Key Name:  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
              Primary Key Id:  ^^^^
   Duplicates Allowed? (Y/N):  ^

   Any Secondary Keys? (Y/N):  ^
Tags Included in Lines? (Y/N):   ^
           Access Type (R/S):  ^
        Number of Data Lines:  ^^^^^^^^
      Number of Primary Keys:  ^^^^^^^^
```

Respond to the prompts as follows:

File Name
: Enter a file name from 1 to 30 characters long to define a file. Names must be unique throughout the dictionary file.

File Id
: Enter a name from 1 to 4 characters long to uniquely identify the file to the dictionary file. The first character must be alphabetic. The remaining characters can be alphabetic or numeric. If you leave this field blank, the data librarian will assign the file ID.

File Description
: The file description is an optional documentation feature. You can store any textual description of the data in this field. This is an excellent means of internal documentation. The file description is included in detailed report listings.

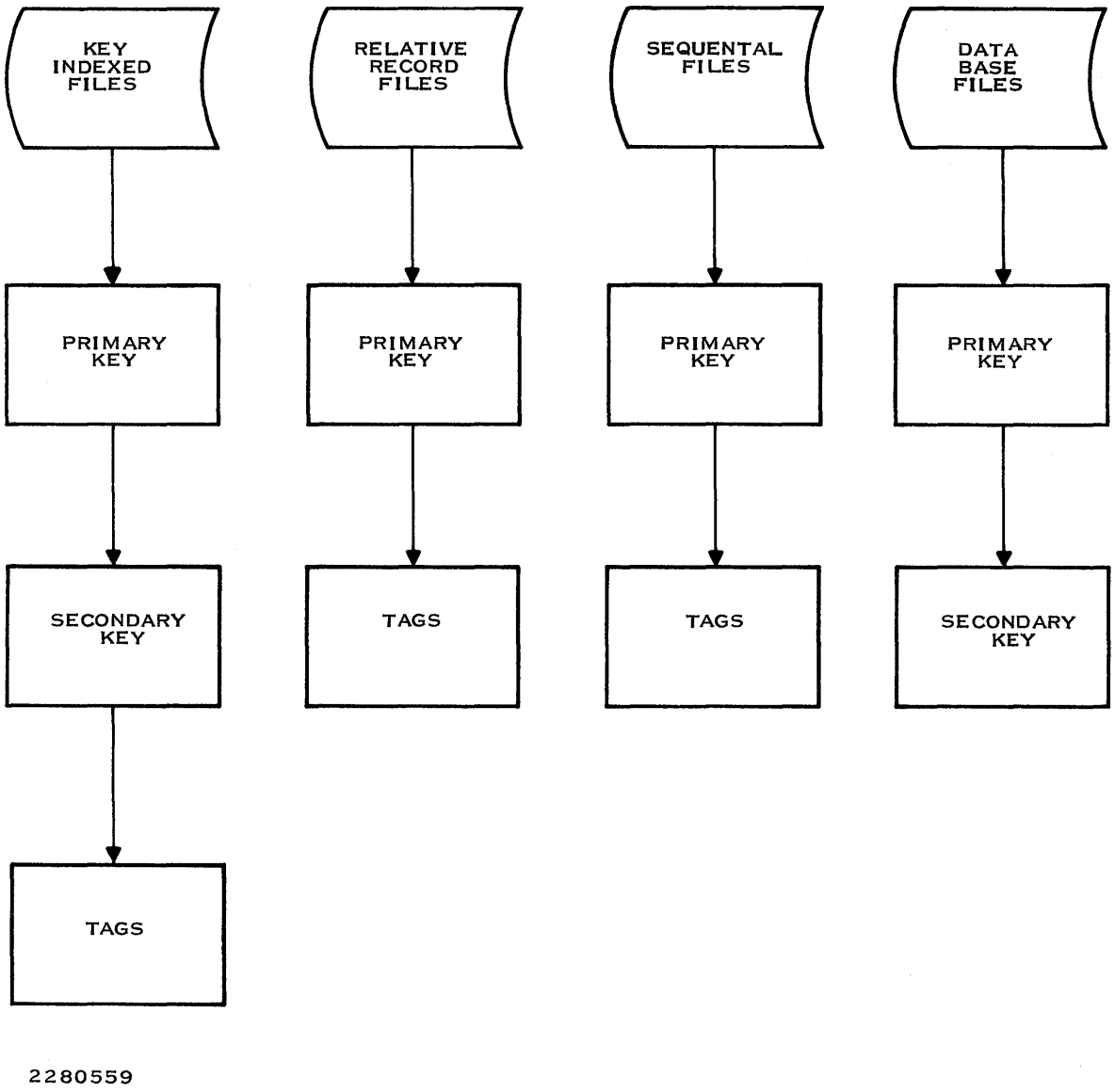


Figure 4-2. Secondary Key and Tag Options for Files

File Type

Enter a two-character code to designate the type of file to be established in the dictionary file. When the cursor is positioned at this prompt, the list of available file types appears at the bottom of the screen. The file types are as follows:

DB — Data base file
KF — Key indexed file
RR — Relative record file
SQ — Sequential file

Primary Key Name

The primary key name is required for all file types. The primary key must be a field for data base, relative record, and sequential files. Only KIFs can also use a group for a primary key.

For relative record and sequential files the record number is defined as the primary key. For these files a default key ID is supplied. To select the default, tab through the primary key name and ID field.

For KIFs, the primary key must be defined in the same position in each line in the file definition.

In response to the Primary Key Name prompt, enter the name. The data librarian verifies that the name exists in the dictionary file and supplies the ID.

Primary Key Id

The primary key ID is required for all file types. If the name has been specified, the ID is supplied by the data librarian; otherwise you should enter the ID. The data librarian verifies that the ID exists in the dictionary file and supplies the name.

Duplicates Allowed?(Y/N)

This prompt asks if duplicate primary keys are allowed. This applies only to KIFs.

Any Secondary Keys?(Y/N)

This prompt applies only to data base files and KIFs. The secondary key provides direct access to line-level data. These secondary keys are included on the line screens for the file.

Tags Included in Lines?(Y/N)

Tags can be included only for relative record files, sequential files, and KIFs. Tags identify unique line types within a conventional file.

Access Type(R/S)

Data base files are the only files that require either random or sequential access options.

Number of Data Lines

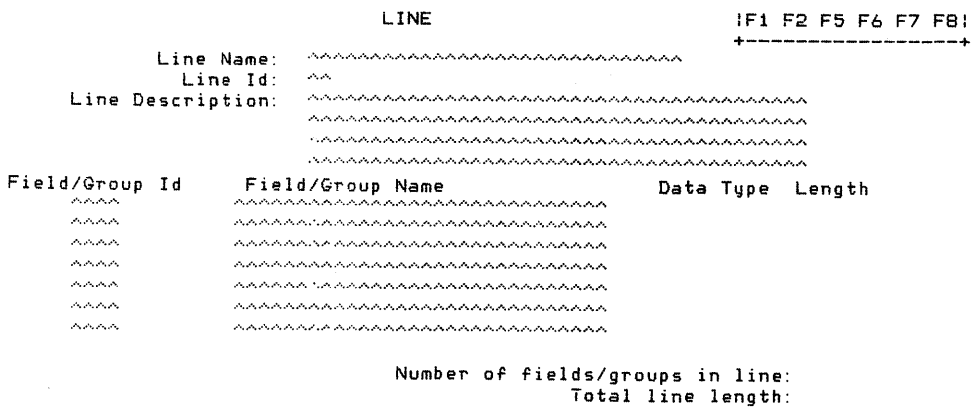
The number of data lines is required only for data base files.

Number of Primary Keys

The total number of primary keys in the file applies only to data base files.

The file screen contains line, tag, and secondary key options requiring additional screens for their definitions. The necessary screens appear if the appropriate options have been selected on the file screen. They are the line, secondary keys, and tag screens.

4.2.7.1 Line Screen. Once the original file screen has been completed, the following screen appears for lines:



Respond to the prompts as follows:

Line Name
: Enter a name from 1 to 30 characters long to describe a line. Names must be unique throughout the dictionary file.

Line Id
: Enter a two-character value to distinguish line types in a file. The line ID must be unique within the file only.

Line Description
: The line description is an optional documentation feature. This description appears in the detail report listing.

  The names or IDs of all fields or groups in this line type must be entered in the same order in which they exist in the file. The data librarian calculates their offsets accordingly. Fields must be defined in the same order in which they physically appear in order for the data librarian to function properly.

During the line definition screen, the data librarian displays the data type and length for each field or group based on information previously defined in the dictionary file.

Line Operations. After ENTER has been pressed, one of the following prompts appears:

ADD THIS LINE?(Y/N):

UPDATE THIS LINE?(Y/N):

Respond to the prompts as follows:

ADD THIS LINE?(Y/N)
: If the information is new to the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file.

UPDATE THIS LINE?(Y/N)
If the information is already contained in the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

The F7 key allows you to delete a line from an existing file definition. When the F7 key is pressed, the following prompt appears:

DELETE THIS LINE?(Y/N):

Respond to the prompt as follows:

DELETE THIS LINE?(Y/N)
Enter Y if the line is to be deleted. The F7 key can only be used when modifying an existing file definition.

Line Function Keys. The following paragraphs describe the line function keys.

F1 Key. This key rolls forward to the next screen.

F2 Key. This key rolls backward to the previous screen.

F5 Key. This key allows the scroll up function for the field/group information on the lower half of the screen.

F6 Key. This key allows the scroll down function for the field/group information on the lower half of the screen.

F7 Key. When the cursor is positioned on the line name or line ID, this key deletes a line from an existing file definition. When the cursor is positioned on a field or group ID or name, pressing the F7 key deletes that ID from the line definition.

F8 Key. This key inserts fields or groups at a specific position within a line definition. To abort an insert, press the F7 key.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. The line information is not sent to the dictionary file. Enter N to return to the original screen. The entire file definition cannot be aborted from this screen. You must use the back field key to return to the line name and press the F2 key to return to the file screen and press the CMD key in order to abort the file definition.

When you complete the line definition, press the ENTER key to add the information in the dictionary file. The screen is then ready to accept another line definition. Pressing ENTER on a blank line screen terminates the line definition.

4.2.7.2 Data Base Secondary Key Screen. If you selected the secondary key option in the previous file screen for data base files, the following screen appears once all lines have been defined:

```
                          Secondary Keys                    !F1 F2 F5 F6 F7   !
                        for DB file                         +-----------------+

                Enter id from the list below: ^^^^
                             Number of keys: ^^^^^^^^
                               Access (R/S): R

Line     Field/Group     Field/Group Name                   Data    Length  Offset
 Id         Id                                             Type
 --       ----      ------------------------------         --      ----    ----
```

Respond to the prompts as follows:

Enter id from the list below
: The list of all fields and groups included in this file appears on the screen. Enter the ID of the field or group to be designated as a secondary key. You can specify up to 13 secondary keys (one at a time). IDL continues to prompt for field IDs until you press the ENTER key, or until 13 secondary keys have been specified.

Number of keys
: Specify the maximum number of unique values allowed for each key.

Access(R/S)
: Specify sequential or random access.

Data Base Secondary Key Function Keys. The following paragraphs describe the function keys for the data base secondary keys.

F1 Key. This key rolls forward to the next screen.

F2 Key. This key rolls backward to the previous screen.

F5 Key. This key activates the scroll up function for the field/group information on the lower half of the screen.

F6 Key. This key activates the scroll down function for the field/group information on the lower half of the screen.

F7 Key. You can delete a secondary key from the file definition by pressing the F7 key when the cursor is displayed next to the secondary key.

Backfield Key. You can use the backfield key to view the list of secondary keys included in this file.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

```
ABORT THIS SCREEN? (Y/N):
```

Enter Y if the screen is to be aborted. Secondary key information is not sent to the dictionary file. Enter N to return to the original screen. The entire file definition cannot be aborted from this screen. You must use the F2 key to return to the file screen and then press the CMD key in order to abort the file definition.

When you complete the secondary key definition, the IDL includes the information in the dictionary file.

4.2.7.3 KIF Secondary Key Screen. If you included secondary keys for KIFs on the file screen, the following screen appears:

```
                        Secondary Keys                    !F1 F2 F5 F6 F7   !
                      for KIF file                        +-----------------+

            Enter id from the list below: ^^^^
                       Modifiable: (Y/N): ^
                       Duplicates: (Y/N): ^

Line     Field/Group     Field/Group Name                  Data    Length  Offset
 Id         Id                                            Type
 --       ----       ------------------------------       --      ----    ----
```

Respond to the prompts as follows:

Enter id from the list below
: The list of all fields and groups included in this file is displayed on the screen. Enter the ID of the field or group to be designated as a secondary key. You can specify up to 13 secondary keys (one at a time). IDL continues to prompt for field IDs until you press the ENTER key, or until 13 secondary keys have been specified.

Modifiable?(Y/N)
: Secondary keys can be modifiable. A modifiable key is one that you can update with a different value.

Duplicates?(Y/N)
: Secondary keys can have duplicates. Duplicate implies that two or more keys can have same value.

KIF Secondary Key Function Keys. The following paragraphs describe the function keys for the KIF secondary key.

F1 Key. This key rolls forward to the next screen.

F2 Key. This key rolls backward to the previous screen.

F5 Key. This key activates the scroll up function for the field/group information on the lower half of the screen.

F6 Key. This key activates the scroll down function for the field/group information on the lower half of the screen.

F7 Key. To delete a secondary key from the file definition, press the F7 key when the cursor is positioned at that field on the screen.

Backfield Key. You can use the backfield key to view the list of secondary keys included in this file.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN? (Y/N):

Enter Y if the screen is to be aborted. Secondary key information is not sent to the dictionary file. Enter N to return to the original screen. The entire file definition cannot be aborted from this screen. You must use the F2 key to return to the fill screen and then press the CMD key in order to abort the file definition.

When you complete the secondary key definition, the IDL will include the information in the dictionary file.

4.2.7.4 Tag Screen. If you selected the tag option on the file screen, the following screen appears:

```
                Tag Information for file      in line    !F1 F2 F5 F6      !
                                                         +-----------------+

     Enter id from the list below:  ^^^^
                         Tag value:  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Line     Field/Group     Field/Group Name                  Data    Length   Offset
 Id         Id                                            Type
 --        ----      ------------------------------        --      ----     ----
```

Respond to the prompts as follows:

Enter id from the list below
: The list of all fields and groups included in this file are displayed on the screen. The fields and groups displayed are all character and integer type fields and groups in the file. You must have the same number of tags as line types.

Tag value
: You can specify as tag fields only groups or fields defined as IS or CH data types. Enter a field ID from the list displayed at the bottom of the screen. If you enter an incorrect tag ID, the following prompt appears:

INVALID TAG SPECIFICATION

Choose another field for the tag or use the F2 key to roll backward through the line screens to modify the line definitions for the tag specification. IDL continues to prompt for IDs and values until tags have been defined for each line type in the file.

Tag Function Keys. The following paragraphs describe the tag function keys.

F1 Key. This key rolls forward to the next screen.

F2 Key. This key rolls backward to the previous screen.

F5 Key. This key activates the scroll up function for the field/group information on the lower half of the screen.

F6 Key. This key activates the scroll down function for the field/group information on the lower half of the screen.

Backfield Key. You can use this key to view the list of tag fields included in this file. When you use the backfield key on the first line type and tag field defined, IDL reinitializes the list of tags, which allows you to choose another valid tag field.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. Tag field information is not sent to the dictionary file. Enter N to return to the original screen. The entire file definition cannot be aborted from this screen. You must use the F2 key to return to the file screen and then press the CMD key in order to abort the file definition.

When you complete the tag definition, the IDL will include the information in the dictionary file.

4.2.7.5 File Operations. Once the file has been defined with or without secondary keys and tags, the file operations appear on the screen. Depending on whether the information already exists in the dictionary file, one of the following prompts appear once the entire file has been defined:

ADD THIS FILE?(Y/N):

UPDATE THIS FILE?(Y/N):

Respond to the prompts as follows:

ADD THIS FILE?(Y/N)
If the information is new to the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

UPDATE THIS FILE?(Y/N)
If the information is already contained in the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

The F7 key allows you to delete a file that is not referenced elsewhere in the dictionary. When the F7 key is pressed, the following prompt appears:

DELETE THIS FILE?(Y/N):

Respond to the prompt as follows:

DELETE THIS FILE?(Y/N)
Enter Y if the file is to be deleted.

4.2.7.6 File Function Keys. The following paragraphs describe the file function keys.

F1 Key. This key activates a roll up function through the file information. In the case of the file screen, the F1 key serves to roll forward through all screens that apply specifically to the file definition.

F2 Key. This key activates a roll down function through the file information. In the case of the file screen, the F2 key serves to roll backward through all screens that apply specifically to the file definition.

F7 Key. This key activates a delete function for a file and the attributes of a file that are not referenced elsewhere in the dictionary file. In the case of the file definition, the F7 key can delete lines, secondary keys, and tags when those screens are displayed.

CMD Key. When information is on the file screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. Information is not sent to the dictionary file. Enter N to return to the original screen.

In the case of the file definition, a file can be deleted using the CMD key only when the file screen is displayed. When you complete the file definition, the IDL will include the information in the dictionary file. The screen is then ready to accept another file definition. Use the CMD key to return to the original screen.

4.2.8 Pathname Screen

Defining pathnames in the dictionary file allows the association of a pathname with a file ID or name. This pathname is the default when you do not specify the pathname in the Assign File ID (AFID) command. The following screen appears for pathnames:

```
                    PATHNAME                        !F7              !
                                                    +----------------+


File Name: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File Id: ^^^^

 Pathname: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Respond to the prompts as follows:

File Name
: Enter a previously defined file name or ID. The data librarian will verify that the file exists in the dictionary file.

File Id
: Enter an ID that is from 1 to 4 characters long to uniquely identify the file to the dictionary file.

Pathname
Enter a pathname to be associated with the file. The pathname can contain synonyms. They will be resolved when the file is assigned for Query-990 access.

4.2.8.1 Pathname Operations. Depending upon whether the information already exists in the dictionary file, one of the following prompts will appear:

ADD THIS PATHNAME?(Y/N):

UPDATE THIS PATHNAME?(Y/N):

Respond to the prompts as follows:

ADD THIS PATHNAME?(Y/N)
If the information is new to the dictionary file this prompt will appear. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

UPDATE THIS PATHNAME?(Y/N)
If the information is already contained in the dictionary file, this prompt will appear. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the same screen.

The F7 key allows you to delete a pathname from the file definition. When the F7 key is pressed, the following prompt appears:

DELETE THIS PATHNAME?(Y/N):

Respond to the prompt as follows:

DELETE THIS PATHNAME?(Y/N)
Enter Y if the pathname is to be deleted.

4.2.8.2 Pathname Function Keys. The following paragraphs describe the pathname function keys.

F7 Key. To delete a pathname from the file definition, press the F7 key when that pathname is displayed on the screen.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. Information is not sent to the dictionary file. Enter N to return to the original screen.

When you complete the pathname definition, the IDL will include the information in the dictionary file. The screen is then ready to accept another pathname definition. Use the CMD key to return to the main screen.

4.2.9 Program Screen

The following screen appears for programs:

```
                          PROGRAM                          !F7              !
                                                           +----------------+
                Program Name: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                  Program Id: ^^^^
         Program Description: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                              ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^


                Host Language: ^^^^^^^^


                    File Name: ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                      File Id: ^^^^
```

Respond to the prompts as follows:

Program Name
: Enter a name from 1 to 30 characters long to describe a program. Program names must be unique throughout the dictionary file.

Program Id
: Enter a name from 1 to 4 characters long to uniquely identify the program to the dictionary file. The first character must be alphabetic. The remaining characters can be alphabetic or numeric. If you leave this field blank, a unique ID will be assigned by the data librarian.

Program Description
: The program description is an optional documentation feature. You can store any textual description of the data in this field. This is an excellent means of internal documentation. The program description is included in detailed report listings.

Host Language
: The host language is an optional documentation feature. This information is included in the DD-990 detail and summary reports.

File Name/File Id
: This field is the only place to internally identify the connection between files and programs. This information is essential to cross-reference reports. Enter either the name or the ID of the file accessed by the program. You can enter more than one file.

4.2.9.1 Program Operations. Depending on whether the information already exists in the dictionary file, one of the following prompts appears:

```
ADD THIS PROGRAM?(Y/N):

UPDATE THIS PROGRAM?(Y/N):
```

Respond to the prompts as follows:

ADD THIS PROGRAM?(Y/N)
If the information is new to the dictionary file this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the original screen.

UPDATE THIS PROGRAM?(Y/N)
If the information is already contained in the dictionary file, this prompt appears. Enter Y if the information is to be sent to the dictionary file. Enter N to return to the same screen.

The F7 key allows you to delete a program definition and a file referenced in the program definition from the dictionary file. If the cursor is positioned at the program name on the screen, the entire program definition is deleted. If the cursor is positioned at the file name, the file is deleted from the program definition. When the cursor is positioned at the program name, and the F7 key is pressed, the following prompt appears:

DELETE THIS PROGRAM?(Y/N):

Respond to the prompt as follows:

DELETE THIS PROGRAM?(Y/N)
Enter Y if the program is to be deleted.

4.2.9.2 Program Function Keys. The following paragraphs describe the program function keys.

F7 Key. You can delete a file from the program definition by pressing the F7 key when the cursor is displayed next to the file.

CMD Key. When information is on the screen and you press the CMD key, the following prompt appears:

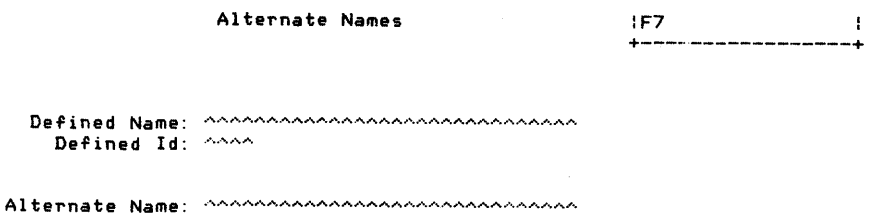
ABORT THIS SCREEN?(Y/N):

Enter Y if the screen is to be aborted. Information is not sent to the dictionary file. Enter N to return to the original screen.

When you complete the program definition, the IDL will include the information in the dictionary file. The screen is then ready to accept another program definition. Use the CMD key to return to the main screen.

4.2.10 Alternate Names Screen

An alternate name is from 1 to 30 characters long and designates a previously defined field, group, file, or program. The purpose of the alternate name is to create a meaningful vocabulary for accessing the elements of the dictionary file. When skillfully applied, the alternate name can become an immediate identification for what the element represents. Alternate names can also be used in Query-990 statements. The following screen appears for alternate names:



Respond to the prompts as follows:

Defined Name
: Enter a previously defined name for a field, group, file, or program. If you enter the defined name, the data librarian supplies the defined ID for that name.

Defined Id
: Enter a previously defined ID for fields, groups, files or programs. The data librarian verifies that the defined ID exists in the dictionary file.

Alternate Name
: Enter one or more names. The alternate names have the same characteristics as the defined name. They must be unique to the dictionary file and can consist of up to 30 characters.

4.2.10.1 Alternate Names Operations. If all the names defined are to be added to the dictionary file, press the ENTER key. The following prompt appears:

UPDATE THESE NAMES?(Y/N):

Enter Y if all the names defined on the screen are to be added to the dictionary file. Enter N to return to the original screen.

To delete an alternate name from the dictionary, be sure that the cursor is positioned on that name in the alternate name field. Press the F7 key and the following screen appears:

DELETE THIS NAME?(Y/N):

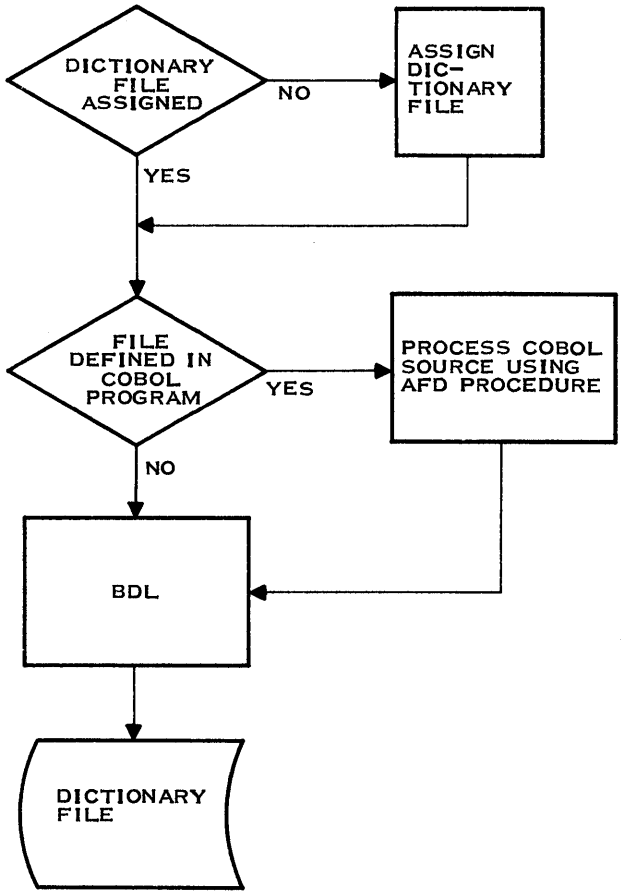
Enter Y if the name entered on the screen is to be deleted. Enter N to return to the original screen.

To abort the process of defining alternate names, press the CMD key while the information is displayed on the screen.

You can use the backfield key to view all the alternate names on the screen.

4.3 BATCH DATA LIBRARIAN — BDL

The BDL uses an input file to add definitions to the dictionary file. The input file must meet the data definition language (DDL) syntax. The DDL syntax consists of rules for combining key words and user-supplied information into the seven kinds of statements required by BDL. The BDL can add definitions to the dictionary file but cannot update or modify them. BDL merely defines a data base file to the dictionary file. A data base file is not created by this definition. You can use the text editor to create files in the proper format to be used by the BDL. Figure 4-3 illustrates the relationships of files to the BDL.



2280560

Figure 4-3. Batch Data Librarian

To execute the data librarian in batch mode, enter the BDL command. The following screen appears:

```
BATCH DATA LIBRARIAN <VERSION L.V.E. YYDDD>

                    PASSWORD:
               INPUT PATHNAME:
        LISTING ACCESS NAME:
              GENERATE IDS?:
                 MODE (F,B):
```

Respond to the prompts as follows:

PASSWORD
: This prompt appears only if DBMS has been installed with security on the system. Enter a previously assigned master password.

INPUT PATHNAME
: Enter the pathname where the input file to be used by BDL is stored.

LISTING ACCESS NAME
: Enter the name of the output file that will receive a copy of the input and any syntax errors that occur.

GENERATE IDS?
: Enter Y if you want BDL to generate unique IDs. When you answer Y to this prompt, BDL replaces any four character ID specified as ???? in the DDL with a unique ID beginning with A001. The primary key ID should be specified as ?001 each time it appears in the DDL. Secondary key IDs should be specified as ?002 through ?013. With this option it is possible to create DDL file descriptions using only long names. The following example illustrates ID generation using BDL.

  You must enter Y if the input to BDL was generated by the AFD procedure.

MODE (F,B)
: Enter B if the BDL is to execute in background mode. Enter F if the BDL is to execute in foreground mode.

```
 BDL INPUT

FILE=????, TYPE=KIF, NAME=DD-PROBLEMS
ID=?001=CN/3.0, DUP=N, MOD=N, NAME=PROBLEM-NUMBER
*
LINE=HE,              NAME=HEADER-INFORMATION
  FIELD=?001=CN/3.0, NAME=PROBLEM-NUMBER
  GROUP=????,         NAME=HEADER-GROUP
    FIELD=?002=CH/6,    NAME=PROBLEM-STATUS
    FIELD=?003=CH/6,    NAME=DD_COMPONENT
    FIELD=????=CH/6,    NAME=PRIORITY
    FIELD=????=CN/6.0, NAME=REPORTED-DATE
    FIELD=????=CH/20,   NAME=REPORTED-BY
    FIELD=????=CN/6.0, NAME=FIXED-DATE
    FIELD=????=CH/20,   NAME=FIXED-BY
  ENDG
ENDL
*
SECONDARY-REFERENCES
?002, DUP=N, MOD=N
?003, DUP=Y, MOD=Y
END.

BDL OUTPUT

BATCH DATA LIBRARIAN   L.V.E.    YYDDD

FILE=A001, TYPE=KIF, NAME=DD-PROBLEMS
ID=A002=CN/3.0, DUP=N, MOD=N, NAME=PROBLEM-NUMBER
*
LINE=HE,              NAME=HEADER-INFORMATION
  FIELD=A002=CN/3.2, NAME=PROBLEM-NUMBER
  GROUP=A003,         NAME=HEADER-GROUP
    FIELD=A004=CH/6,    NAME=PROBLEM-STATUS
    FIELD=A005=CH/6,    NAME=DD_COMPONENT
    FIELD=A006=CH/6,    NAME=PRIORITY
    FIELD_A007=CN/6.0, NAME=REPORTED-DATE
    FIELD=A008=CH/20,   NAME=REPORTED-BY
    FIELD=A009=CN/6.0, NAME=FIXED-DATE
    FIELD=A010=CH/20,   NAME=FIXED-BY
  ENDG
ENDL
*
SECONDARY-REFERENCES
A004, DUP=N, MOD=N
A005, DUP=Y, MOD=Y
END.
```

4.4 DD-990 DDL STATEMENTS

In order for the BDL to process file descriptions for the dictionary file, the file descriptions must conform to the DD-990 DDL syntax. The key words are as follows:

ACCESS	LINES
DESCRIPTION	MOD
DUP	NAME
END.	RANDOM
ENDD	SECONDARY-REFERENCES
ENDG	SEQUENTIAL
ENDK	
ENDL	TAG
FIELD	TYPE
FILE	VALUE
GROUP	VOL
ID	
LINE	

The seven DD-990 DDL statements are as follows:

- FILE
- ID
- LINE
- FIELD
- GROUP
- SECONDARY REFERENCES
- END

4.4.1 Optional DD-990 DDL Features

Three optional DD-990 DDL features allow for internal documentation. They are as follows:

- NAME
- DESCRIPTION
- Comments

Only a few syntax differences occur between conventional files and data base files. TAG, DUP, MOD, ENDK, and VALUE are not allowed for data base files. Also, DD-990 allows NAME and DESCRIPTION for data base files, but DBMS-990 does not.

4.4.2 FILE Statement

The first line of the DD-990 DDL is the FILE statement. The FILE statement format for KIFs, sequential files, and relative record files is as follows:

FILE statement syntax:

FILE= <id>,TYPE=<KIF|SQ|RR>,[TAG]

Example of a FILE statement:

FILE=DF14, TYPE=KIF, TAG

The following describes the components of the FILE statement:

FILE
: The file ID is a unique name from 1 to 4 characters long that identifies the file to the dictionary file. The first character of the unique name must be alphabetic. The remaining characters can be either alphabetic or numeric.

TYPE
: The file type is KIF for key indexed files, SEQ for sequential files, and RR for relative record files.

TAG
: Tag indicates that tag field values are defined for each line in the file. If you do not specify TAG, the file cannot contain tags.

4.4.3 ID Statement

Descriptions of the ID statement format are as follows:

ID statement syntax:

ID = <id>=<data format>,[DUP={y|n}]

Example of an ID statement:

ID=PRKY=IS, DUP=Y

The following describes the components of the ID statement:

ID
: The ID is from 1 to 4 characters long and identifies the primary key. The ID is used to access the file through the primary key.

<data format>

The data format describes the data type of the primary key. Three types of formats are available. Type one format specifies the format code. Type two format specifies the format code and the field length. Type three format specifies the format code, field length, and number of decimal places. The data types of primary keys must be ID for relative record and sequential files. See Appendix B for detailed listings of the data types supported by DD-990. Examples of the three format codes are as follows:

TYPE ONE:	IS
TYPE TWO:	CH/5
TYPE THREE:	CN/5.1

DUP

DUP signifies whether duplicate values are allowed for the primary key. DUP applies only to KIFs.

MOD

MOD specifies whether the primary key is modifiable and applies only to KIFs.

4.4.4 LINE Statement

The LINE statement format for conventional files is as follows:

LINE statement syntax:

```
LINE = <line type>
.
. <field and group statements>
.
ENDL
```

Example of a LINE statement:

```
LINE = 08
.
.
.
ENDL
```

The following describes the components of the LINE statement:

LINE

The line type is a two-character ID and can be alphabetic or numeric. The line type must be unique within the file definition.

ENDL

The END LINE statement indicates the end of a line specification.

4.4.5 FIELD Statement

FIELD statements are allowed only between LINE and ENDL, or GROUP and ENDG. The FIELD statement format for conventional files is as follows:

FIELD statement syntax:

FIELD = <field id> = <data format>,VALUE = <"tag field value">

Example of a FIELD statement:

FIELD = AB02 = IS,VALUE = "03'6"

The following describes the components of the FIELD statement:

FIELD
: The field ID identifies the field. The field ID is a unique name from 1 to 4 characters long that identifies the field to the dictionary file. The <data format> describes the data type of the field.

VALUE
: The value associated with the field defines the tag value. You must use single or double quotes to contain the tag value. To insert space in this value, enclose the blanks in whichever kind of quote (single or double) has not been used to contain the tag value. Quotes are optional for integers.

4.4.6 GROUP Statement

The GROUP and ENDG statements bracket a list of fields (or a single field). You cannot define groups within other groups. The GROUP statement format for conventional files is as follows.

GROUP statement syntax:

```
GROUP = <group id>
.
.
.
ENDG
```

Example of a GROUP statement:

```
GROUP = AB05
FIELD = AB02 = IS
FIELD = AB03 = CH/2
ENDG
```

The following describes the components of the GROUP statement:

GROUP
: The group ID identifies the group. The group ID is a unique name from 1 to 4 characters long that identifies the group to the data dictionary file. You can declare the group as a secondary key. Each GROUP statement requires at least one FIELD statement.

ENDG
The ENDG statement defines the end of a group. Any succeeding FIELD statements are not part of the preceding group definition. Succeeding group definitions are allowed.

4.4.7 SECONDARY-REFERENCES Statement

Only one SECONDARY-REFERENCES statement is allowed in the DDL for a file. The SECONDARY-REFERENCES statement allows you to define secondary keys in KIFs. Secondary keys are allowed for KIFs only. Up to thirteen secondary keys are allowed. The syntax for defining secondary keys is as follows:

SECONDARY-REFERENCES statement syntax:

SECONDARY-REFERENCES
{<group id>|<field id>},[DUP = {y|n}],[MOD = {y|n}]

Example of a SECONDARY-REFERENCES statement:

ADDR, DUP = N, MOD = Y

The following describes the components of the SECONDARY-REFERENCES statement:

GROUP ID
The group ID is from 1 to 4 characters long and identifies the group to the dictionary file.

FIELD ID
The field ID is from 1 to 4 characters long and identifies the field to the dictionary file.

DUP
Dup specifies whether duplicate values are allowed for the key.

MOD
MOD specifies whether the secondary key is modifiable.

4.4.8 END. Statement

The END. statement is the last line of the DD-990 DDL. The END. statement format is as follows:

END. statement syntax:

END.

The following describes the END. statement component:

END.
The END. statement defines the end of the file definition.

4.4.9 NAME syntax

The NAME feature can follow a FILE, ID, LINE, GROUP, or FIELD statement. This feature is not allowed in the SECONDARY KEY statement. NAME provides an alias capability. It allows entities to be referenced by something more meaningful than a four-character ID. The syntax for NAME is as follows:

```
NAME=<name>
```

The following describes the component of the NAME feature:

NAME
: A NAME can be up to 30 characters long and must begin with a letter. The rest of the NAME can consist of any combination of letters, digits, dashes, underscores, or dollar signs.

4.4.10 DESCRIPTION Syntax

DESCRIPTION may be defined for files, lines, groups, or fields. You cannot use the DESCRIPTION feature in SECONDARY KEY statements. The syntax for DESCRIPTION is as follows:

```
DESCRIPTION
<descriptive text>
ENDD
```

The following describes the components of the DESCRIPTION feature:

DESCRIPTION
: DESCRIPTION marks the beginning of descriptive text.

<descriptive text>
: The descriptive text can be up to 160 characters long.

ENDD
: The ENDD statement indicates the end of the description specification.

4.4.11 Comments

Any information following an asterisk (*) is considered a comment. Some examples of comments are as follows:

```
*This is a comment before a statement
GROUP=<group id>
*This is a comment after a statement
ENDG *This is a comment on the same line
```

4.5 SYNTAX EXAMPLES

The following paragraphs provide examples of sequential, relative record, and key indexed files.

4.5.1 Sequential File Syntax

The total of all field lengths within a line should match the logical record length of the sequential file. The following example represents the DD-990 DDL for a sequential file, CUST:

```
FILE=CUST, TYPE=SQ
NAME=CUSTOMER-FILE
*
ID=CSTN=ID/4,  NAME=CUSTOMER-NUMBER
*
LINE=CS
  FIELD=NAME=CH/20,  NAME=CUSTOMER-NAME
  GROUP=ADDR,  NAME=CUSTOMER-ADDRESS
    FIELD=STRE=CH/20,  NAME=STREET
    FIELD=CITY=CH/15
    FIELD=STAT=CH/2
    FIELD=ZIP=CN/5.0
  ENDG
ENDL
END.
```

4.5.2 Relative Record File Syntax

The total of all field lengths within a line should match the logical record length of the relative record file. The following example represents the DD-990 DDL for a relative record file, INVT:

```
FILE=INVT, TYPE=RR
NAME=INVENTORY-FILE
*
ID=PTNO=ID/4
NAME=PART-NUMBER
*
LINE=01
  FIELD=DESC=CH/15
  NAME=ITEM-DESCRIPTION
  FIELD=PRIC=CN/6.2
  NAME=UNIT-PRICE
  FIELD=QUOH=CN/5.0
  NAME=QUANTITY-ON-HAND
ENDL
END.
```

4.5.3 Key Indexed File Syntax

The total of all field lengths within a line should match the logical record length of the key indexed file. Primary and secondary keys defined in the DD-990 DDL must start at the same offset as the keys defined for the file when created. The following example represents the DD-990 DDL for a KIF, INVO:

```
FILE=INVO, TYPE=KIF, TAG
NAME=INVOICE-FILE
DESCRIPTION
 For each customer order, there is one HE
 line containing header information for
 the invoice. In addition, there will
 be one IT line for each item ordered.
ENDD
*
ID=INUM=CN/10.0, DUP=Y
NAME=INVOICE-NUMBER
*
LINE=HE, NAME=HEADER-INFORMATION
  FIELD=TAG1=CH/2, VALUE=01
  FIELD=INUM=CN/10.0
  GROUP=DATE, NAME=DATE-OF-PURCHASE
    FIELD=MNTH=CN/2.0, NAME=MONTH
    FIELD=DAY=CN/2.0
    FIELD=YEAR=CN/2.0
  ENDG
  FIELD=BUYR=ID/4, NAME=BUYER
ENDL
*
LINE=IT, NAME=ITEM-LINE
  FIELD=TAG2=CH/2, VALUE=02
  FIELD=INUM=CN/10.0
  FIELD=PART=ID/4, NAME=PART-NUMBER-ORDERED
  FIELD=QUAN=CN/5.0, NAME=QUANTITY-ORDERED
ENDL
*
SECONDARY-REFERENCES
PART, DUP=Y, MOD=N
END.
```

4.5.4 Primary Key as a Group

The primary key can be a group as well as a field for KIF. The example in Figure 4-4 represents a file using a group as the primary key.

```
FILE=ACTV, TYPE=KIF
NAME=LABOR-MATERIAL-ACTIVITY
DESCRIPTION
LABOR/MATERIAL ACTIVITY CHARGED FILE
ENDD
*
ID=PKEY=GROUP, DUP=N, MOD=N
NAME=ACTIVITY-PRIMARY-KEY
  FIELD=JOB=CN/5.0,  NAME=JOB-NUMBER
  FIELD=ACT=CN/3.0,  NAME=ACTIVITY
ENDK
*
LINE=AA
DESCRIPTION
LABOR/MATERIAL ACTIVITY CHARGED RECORD
ENDD
  GROUP=PKEY
    FIELD=JOB=CN/5.0
    FIELD=ACT=CN/3.0
  ENDG
  FIELD=ESTH=PK/8.1,   NAME=EST-HOURS
  FIELD=CURH=PK/8.1,   NAME=CURRENT-HOURS
  FIELD=TODH=PK/12.2,  NAME=TODATE-HOURS
  FIELD=PERH=CN/4.1,   NAME=PERCENT-LABOR-COMPLETED
  FIELD=ESTM=PK/12.2,  NAME=EST-MATERIAL
  FIELD=CURM=PK/12.2,  NAME=CURRENT-MATERIAL
  FIELD=TODM=PK/12.2,  NAME=TODATE-MATERIAL
  FIELD=PERM=CN/4.1,   NAME=PERCENT-MATERIAL
ENDL
*
END.
```

Figure 4-4. Primary Key as Group

4.6 DATA BASE FILE SYNTAX

The DDL for data base files follows the syntax described earlier with four exceptions. The exceptions for data base files are as follows:

- The FILE statement must have the following form:

  FILE = <id>,LINES = <file size>

- The <data format> in the primary key definition is written as follows:

  PRIMARY KEY DEFINITION
  ID = <id> = <data format>[,ACCESS = {random/1|sequential/1}]

- The ID in a SECONDARY-REFERENCES statement is written as follows:

  SECONDARY-REFERENCES
  {group id|field id} = VOL = <number of different values of this key>
  .
  .
  .

- The following key words are not allowed for data base files:

  TAG
  VALUE
  DUP
  MOD
  ENDK

The example in Figure 4-5 represents the DDL for a data base file. A complete description of the DDL for data base files is found in the *Data Base Administrator User's Guide*.

NOTE

The Batch Data Librarian (BDL) merely defines a data base file to the dictionary file. A data base file is not created by this definition.

```
FILE=SOFL, LINES=704,  NAME=SALES_ORDER_FILE
DESCRIPTION
THIS IS THE SALES ORDER FILE FROM WHICH INVOICES ARE PRODUCED
EACH MONTH.
ENDD
*
ID=SONM=CH/6, VOL=50, ACCESS=RANDOM/1
NAME=SALES_ORDER_NUMBER
*
LINE=BL
FIELD=BILL=CH/5,  NAME=BILL_TO_CUSTOMER
FIELD=LOCK=CH/2
ENDL
*
LINE=02
FIELD=SHIP=CH/5,  NAME=SHIP_TO_CUSTOMER
ENDL
*
LINE=03
FIELD=ITEM=CH/4,   NAME=ITEM_ORDERED

FIELD=QUAN=CN/4.0,   NAME=QUANTITY_ORDERED
ENDL
*
SECONDARY-REFERENCES
BILL=VOL=50, ACCESS=SEQUENTIAL/1
END.
```

Figure 4-5. DDL Syntax for Data Base File

5

DD-990 Utilities

5.1 INTRODUCTION

DD-990 has a group of utilities designed to produce reports that list the contents and status of the definitions in the dictionary file. Reports showing the relationships between the different entities and attributes can be valuable decision-making tools for expanding organizations. The DD-990 report utilities are as follows:

- AFD — Generates DDL files from COBOL source programs suitable for entry into the dictionary file through BDL
- DDR — Generates summary and detail reports of the dictionary file contents
- DDXREF — Generates cross-reference reports of the dictionary file contents
- LSTFIL — Generates a listing of a file definition from the dictionary file in DDL format.
- GCB — Generates COBOL record discriptions for use in the file definition section of COBOL programs

You cannot use the report utilities for DD-990 unless the Assign Data Dictionary (ADD) command was executed after one of the following situations:

- A system boot
- The RDD command
- The Start Data Base command (in a DBMS environment)

Refer to Section 3 for more information on the ADD procedure.

5.2 AUTOMATIC FILE DEFINITION — AFD

The AFD utility generates DDL files from the file definition section of COBOL source programs. These DDL files are in proper format and syntax to be included in the dictionary file through the BDL. The following screen appears for AFD:

```
AUTOMATIC FILE DEFINITION <VERSION L.V.E. YYDDD>
                COBOL SOURCE:
                    FILE NAME:
                  DDL SOURCE:
```

Respond to the prompts as follows:

COBOL SOURCE
Enter the pathname of the input COBOL source program containing the file to be defined.

FILE NAME
Enter the name of the file specified in the FD statement within the COBOL source program.

DDL SOURCE
Enter the pathname of the output file that will contain the new DDL source.

The AFD utility generates DDL from COBOL source programs. Figure 5-1 shows an example of an input file to be used by the AFD utility.

After the AFD utility has processed definitions from COBOL source programs, a listing of the output is generated. The following screen generates the example of output from the AFD Utility as shown in Figure 5-2. This particular example is the same information contained in the LSTFIL output example.

```
AUTOMATIC FILE DEFINITION  <L.V.E.  YYDDD>

                COBOL SOURCE:  <pathname of COBOL source program>
                    FILE NAME:  PAYMENTS-RECEIVED-FILE
                  DDL SOURCE:  <pathname of output file>
```

Any COBOL syntax that cannot be directly translated to DDL will be flagged as a warning or as an error. Warnings reflect COBOL syntax that has been commented out in the DDL, yet does not make the AFD output unacceptable to the BDL. Fatal errors reflect COBOL syntax that cannot be translated to DDL or commented out. Not all COBOL constructs are handled. Refer to paragraph 8.8 for AFD error messages. A DDL with fatal errors will not be successfully processed by the BDL. Warnings or fatal errors can be resolved by modifying the AFD output using the text editor so that it becomes legal DDL syntax.

Once any warnings or fatal errors have been resolved, this file can be used by the BDL. The BDL generates unique IDs for all unresolved entity names (represented by question marks in the AFD output). The text editor can be used to generate four character IDs and/or add descriptions for the entities. BDL can generate four character IDs automatically.

```
IDENTIFICATION DIVISION.

PROGRAM-ID.      PAYMENTS.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER.  TI-990.
OBJECT-COMPUTER.  TI-990.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
     SELECT PAYMENTS-RECEIVED-FILE
          ASSIGN TO RANDOM,
               "COMP61.ACCT.PAYMENT";
          ORGANIZATION IS INDEXED;
          ACCESS IS DYNAMIC;
          RECORD KEY IS P-CUST-NUMBER;
          ALTERNATE RECORD KEY IS P-EXPIRES-DATE
               WITH DUPLICATES;
          ALTERNATE RECORD KEY IS P-ORDER-NUM1
               WITH DUPLICATES;
          ALTERNATE RECORD KEY IS P-ORDER-NUM2
               WITH DUPLICATES;
          FILE STATUS IS PAYMENTS-FILE-STATUS.
DATA DIVISION.

FILE SECTION.
FD   PAYMENTS-RECEIVED-FILE
     LABEL RECORD IS STANDARD
     RECORD CONTAINS 96 CHARACTERS
     BLOCK CONTAINS 2592 CHARACTERS
     DATA RECORD IS PAYMENTS-RECORD.
01   PAYMENTS-RECORD.
     02   P-CUST-NUMBER          PIC 9(8).
     02   P-EXPIRES-DATE.
          03   P-EXP-DATE        PIC 9(2)          COMP-1.
     02   P-TERMS-AGREED         PIC S9(2).
     02   P-CONT-AMT             PIC 9(6)V99.
     02   P-AMOUNT-PAID1         PIC S9(5)V99      COMP-3.
     02   P-TYPE-PAYMENT1        PIC X(1).
     02   P-AMOUNT-PAID2         PIC S9(5)V99      COMP-3.
     02   P-TYPE-PAYMENT2        PIC X(1).
     02   P-ORDER-NUM1           PIC X(10).
     02   P-ORDER-NUM2           PIC X(10).
     02   FILLER                 PIC X(5).
WORKING-STORAGE SECTION.
```

Figure 5-1. AFD Input File

```
*
*          PROGRAM NAME = PAYMENTS
*
*          DDL FOR FILE =PAYMENTS-RECEIVED-FILE
*
*          GENERATED BY AUTOMATIC FILE DEFINITION UTILITY
*               AT HH:MM:SS  DD/MM/YY
*
 FILE=????, TYPE=KIF, NAME=PAYMENTS-RECEIVED-FILE
 ID=?001=CN/8.0, DUP=NO, MOD=NO
 LINE=01, NAME=PAYMENTS-RECORD
 FIELD=?001=CN/8.0, NAME=P-CUST-NUMBER
 GROUP=?002, NAME=P-EXPIRES-DATE
    FIELD_????=IS/2, NAME=P-EXP-DATE
 ENDG
 FIELD=????=CS/3.0, NAME=P-TERMS-AGREED
 FIELD=????=CN/8.2, NAME=P-CONT-AMT
 FIELD=????=PK/8.2, NAME=P-AMOUNT-PAID1
 FIELD=????=CH/1, NAME=P-TYPE-PAYMENT1
 FIELD=????=PK/8.2, NAME=P-AMOUNT-PAID2
 FIELD=????=CH/1, NAME=P-TYPE-PAYMENT2
 FIELD=?003=CH/10, NAME=P-ORDER-NUM1
 FIELD=?004=CH/10, NAME=P-ORDER-NUM2
 FIELD=????CH/5
 ENDL
 SECONDARY-REFERENCES
 ?002, DUP=YES, MOD=YES
 ?003, DUP=YES, MOD=YES
 ?004, DUP=YES, MOD=YES
 END.
*
*
*>>---> "PAYMENT-RECEIVED-FILE" HAS NO WARNINGS AND NO FATAL ERRORS
```

Figure 5-2. AFD Output

5.3 DATA DICTIONARY REPORTS — DDR

The DDR utility generates detail and summary reports of all the contents of the dictionary file. There are several reporting options for the DDR utility. They are as follows:

- Specify only the NAME or ID to report on a specific entity occurrence
- Specify the ENTITY TYPE but not the ENTITY NAME/ID to report on all of one type
- Specify ENTITY TYPES separated by commas (,) without specifying NAME/ID to report on a combination of types
- Leave ENTITY TYPE and NAME/ID blank to report on the entire dictionary file

The following screen appears for DDR:

```
DATA DICTIONARY REPORT GENERATOR <VERSION L.V.E. YYDDD>

                          PASSWORD:
                 DETAIL REPORT(Y/N):
           ENTITY TYPE(PR/FI/GR/FL):
                      ENTITY NAME/ID:
                LISTING ACCESS NAME:
                          MODE(F,B):
```

Respond to the prompts as follows:

PASSWORD
This prompt appears only if DBMS-990 with security is installed on the system. Enter the previously assigned master password.

DETAIL REPORT(Y/N)
Enter Y if the report is to list all the detail information about each entry in the dictionary file. Enter N if the report is to list information in a summary report.

ENTITY TYPE(PR/FI/GR/FL)
Enter the type of entity to be reported from the dictionary file. The different types are as follows:

PR —Program
FI —Field
GR —Group
FL —File

ENTITY NAME/ID
Enter the name or ID of the entity to be accessed from the dictionary file.

LISTING ACCESS NAME
Enter the name of a device where the output is to be sent.

MODE (F,B)
Enter B to execute DDR in background mode. Enter F to execute DDR in foreground mode.

The DDR utility generates reports for all the group definitions in the dictionary file. The following screen generates an example of a summary report on group definitions as shown in Figure 5-3.

```
DATA DICTIONARY REPORT GENERATOR  <L.V.E.  YYDD>

                          PASSWORD:  <master password>
                 DETAIL REPORT (Y/N)?:  NO
           ENTITY TYPE (PR/FI/GR/FL):  GR
                      ENTITY NAME/ID:
                LISTING ACCESS NAME:  <pathname of output file>
                          MODE (F,B):  F
```

```
DD-990 SUMMARY REPORT     L. V. E.   YYDDD                    PAGE 1
GROUPS
  ID     NAME                                    COUNT  LENGTH   CREATED    UPDATED
  ____   ____________________________________    _____  ______   ________   _______
  A004   DATE-IN                                   3      6      07/31/81
    1    A005   MONTH-IN
    1    A006   DAY-IN
    1    A007   YEAR-IN
  ID     NAME                                    COUNT  LENGTH   CREATED    UPDATED
  ____   ____________________________________    _____  ______   ________   _______
  A016   P-EXPIRES-DATE                           1      2      07/31/81
    1    A017   P-EXP-DATE
  ID     NAME                                    COUNT  LENGTH   CREATED    UPDATED
  ____   ____________________________________    _____  ______   ________   _______
  ADDR   CUSTOMER-ADDRESS                         4      42     07/31/81
    1    STRE   STREET
    1    CITY
    1    STAT
    1    ZIP
  ID     NAME                                    COUNT  LENGTH   CREATED    UPDATED
  ____   ____________________________________    _____  ______   ________   _______
  DATE   DATE-OF-PURCHASE                         3      6      07/31/81
    1    MNTH   MONTH
    1    DAY
    1    YEAR
  ID     NAME                                    COUNT  LENGTH   CREATED    UPDATED
  ____   ____________________________________    _____  ______   ________   _______
  PKEY   ACTIVITY-PRIMARY-KEY                     2      8      07/31/81
    1    JOB    JOB-NUMBER
    1    ACT    ACTIVITY

DD-990 SUMMARY REPORT     L. V. E.   YYDDD
   NAME                                 ID     TYPE
   ___________________________________  ____   ____
   CUSTOMER-ADDRESS                     ADDR   GR
   DATE-IN                              A004   GR
   DATE-OF-PURCHASE                     DATE   GR
   P-EXPIRES-DATE                       A016   GR
   TOTAL NUMBER OF ENTITIES REPORTED =      5
```

Figure 5-3. DD-990 Summary Report

The DDR utility generates reports by detail for all the program definitions in the dictionary file. The following screen generates an example of the detail report on all programs definitions defined in the dictionary file as shown in Figure 5-4.

```
DATA DICTIONARY REPORT GENERATOR  <L. V. E.   YYDDD>

                    PASSWORD:  <master password>
        DETAIL REPORT (Y/N):  YES
ENTITY TYPE (PR/FI/GR/FL):  PR
              ENTITY NAME/ID:
        LISTING ACCESS NAME:  <pathname of output file>
                  MODE(F,B):  F
```

```
DD-990 DETAIL REPORT    L. V. E.    YYDDD
PROGRAMS

                                                          DATE       LAST
ID     NAME                                 LANGUAGE      CREATED    UPDATED
----   ----------------------------------   ----------    --------   --------
A027   INVENTORY-UPDATE                     COBOL         07/31/81

       DESCRIPTION
       -----------
         This program is run once a month to
         adjust inventory for the month's
         activity.

       ID     NAME                                 TYPE
       ----   ----------------------------------   ----
       INVT   INVENTORY-FILE                        RR
       ACTV   LABOR-MATERIAL-ACTIVITY               KF

                                                          DATE       LAST
ID     NAME                                 LANGUAGE      CREATED    UPDATED
----   ----------------------------------   ----------    --------   --------
A028   INCOME-PROGRAM                       COBOL         07/31/81

       DESCRIPTION
       -----------
         This program calculates sales income on
         a monthly basis.

       ID     NAME                                 TYPE
       ----   ----------------------------------   ----
       INVO   INVOICE-FILE                          KF


NAME                                 ID     TYPE
----------------------------------   ----   ----
INCOME-PROGRAM                       A028   PR
INVENTORY-UPDATE                     A027   PR

TOTAL NUMBER OF ENTITIES REPORTED =      2
```

Figure 5-4. DD-990 Detail Report (All Programs)

The DDR utility generates reports on individual entities in the dictionary file. The following screen generates an example of the detail report on the single entity PAYMENTS-RECEIVED-FILE as shown in Figure 5-5.

```
DATA DICTIONARY REPORT GENERATOR  <L. V. E.   YDDDD>

                  PASSWORD: <master password>
      DETAIL REPORT (Y/N): YES
ENTITY TYPE (PR/FI/GR/FL):
           ENTITY NAME/ID: PAYMENTS-RECEIVED-FILE
       LISTING ACCESS NAME: <pathname of output file>
               MODE(F,B): F
```

```
DD-990 DETAIL REPORT   L.V.E.   YYDDD
FILES
                                                    FILE          DATE      LAST
  ID     NAME                                       TYPE          CREATED   UPDATED
  ----   ----------------------------------------   ----          -------   -------
  A014   PAYMENTS-RECEIVED-FILE                     KF            07/31/81

         ID     NAME                                TYPE  FORMAT  OFFSET  ACCESS
         ----   ----------------------------------  ----  ------  ------  ------
(PK)     A015   P-CUST-NUMBER                       CN    8.0     0       SEQUENTIAL

(SK)     A016   P-EXPIRES-DATE                      GR    2       8       SEQUENTIAL
(SK)     A024   P-ORDER-NUM1                        CH    10      31      SEQUENTIAL
(SK)     A025   P-ORDER-NUM2                        CH    10      41      SEQUENTIAL

(LN)     01     PAYMENTS-RECORD
  1      A015   P-CUST-NUMBER                       CN    8.0     0
  1      A016   P-EXPIRES-DATE                      GR    2       8
  2      A017   P-EXP-DATE                          IS    2       8
  1      A018   P-TERMS-ACREED                      CS    3.0     10
  1      A019   P-CONT-AMT                          CN    8.2     13
  1      A020   P-AMOUNT-PAID1                      PK    8.2     21
  1      A021   P-TYPE-PAYMENT1                     CH    1       25
  1      A022   P-AMOUNT-PAID2                      PK    8.2     26
  1      A023   P-TYPE-PAYMENT2                     CH    1       30
  1      A024   P-ORDER-NUM1                        CH    10      31
  1      A025   P-ORDER-NUM2                        CH    10      41
  1      A026                                       CH    5       51

  TOTAL NUMBER OF ENTITIES REPORTED =      1
```

Figure 5-5. DD-990 Detail Report (Single Entity)

5.4 DATA DICTIONARY CROSS-REFERENCE REPORTS — DDXREF

The DDXREF utility generates cross-reference listings that show the relationships between entities in the dictionary file. The following screen appears for DDXREF:

```
DATA DICTIONARY CROSS REFERENCE <VERSION L.V.E. YYDDD>

                    PASSWORD:
                ENTITY NAME/ID:
          LISTING ACCESS NAME:
                   MODE(F,B):
```

Respond to the prompts as follows:

PASSWORD
: This prompt appears only if DBMS-990 security is installed on the system. Enter the previously assigned master password.

ENTITY NAME/ID
: If you do not respond to this prompt, a cross-reference of the entire dictionary file is printed. To print only a cross-reference of a specific entity, enter the specific entity name or ID.

LISTING ACCESS NAME
Enter a pathname where the report will be output.

MODE(F,B)
Enter B to execute DDXREF in background mode. Enter F to execute DDXREF in foreground mode.

The DDXREF utility generates cross-reference reports on individual entities. The following screen generates an example of a cross-reference report on the single entity ZIP, as shown in Figure 5-6.

```
        DATA DICTIONARY CROSS-REFERENCE <L.V.E.   YYDDD>

                  PASSWORD: <master password>
            ENTITY NAME/ID: ZIP
          LISTING ACCESS NAME: <pathname of output file>
                 MODE(F,B): F

DD-990 CROSS REFERENCE REPORT              L.V.E.  YYDDD

   ID     TYPE  NAME                                  USED IN   TYPE  KEY
   ----   ----  ------------------------------        -------   ----  ---
   ZIP    FL                                          CUST      FI
                                                      ADDR      GR

DD-990 CROSS REFERENCE REPORT              L.V.E.  YYDDD

   NAME                              ID
   --------------------------------  ----
   CUSTOMER-ADDRESS                  ADDR
   CUSTOMER-FILE                     CUST


   TOTAL NUMBER OF ENTITIES REPORTED =       1
```

Figure 5-6. DD-990 Cross-Reference Report (Single Entity)

The DDXREF utility generates cross-reference reports on the entire contents of the dictionary file as well as individual entities. The following screen generates an example of a cross-reference report on an entire dictionary file as shown in Figure 5-7.

```
   DATA DICTIONARY CROSS-REFERENCE <L.V.E.   YYDDD>

             PASSWORD: <master password>
       ENTITY NAME/ID:
   LISTING ACCESS NAME: <pathname of output file>
            MODE(F,B): F
```

```
DD-990 CROSS REFERENCE REPORT    L. V. E.    YYDDD
   ID     TYPE  NAME                              USED IN    TYPE  KEY
   ----   ----  --------------------------------  -------    ----  ---
   A000   FL    SEQ-RR-FILES-KEY
   A001   FI    CARD-FILE
   A002   FL                                      A001       FI    PK
   A003   FL    PUNCH                             A001       FI
   A004   GR    DATE-IN                           A001       FI
   A005   FL    MONTH-IN                          A001       FI
                                                  A004       GR
   A006   FL    DAY-IN                            A001       FI
                                                  A004       GR
   A007   FL    YEAR-IN                           A001       FI
                                                  A004       GR
   A008   FL                                      A001       FI
   A009   FL    CD-PUNCH                          A001       FI
   A010   FL    CD-SALESMAN-NBR                   A001       FI
   A011   FL    CD-CUSTOMER-NBR                   A001       FI
   A012   FL    CD-SALES-AMT                      A001       FI
   A013   FL                                      A001       FI
   A014   FI    PAYMENTS-RECEIVED-FILE
   A015   FL    P-CUST-NUMBER                     A014       FI    PK
   A016   GR    P-EXPIRES-DATE                    A014       FI    SK
   A017   FL    P-EXP-DATE                        A014       FI
                                                  A016       GR
   A018   FL    P-TERMS-AGREED                    A014       FI
   A019   FL    P-CONT-AMT                        A014       FI
   A020   FL    P-AMOUNT-PAID1                    A014       FI
   A021   FL    P-TYPE-PAYMENT1                   A014       FI
   A022   FL    P-AMOUNT-PAID2                    A014       FI
   A023   FL    P-TYPE-PAYMENT2                   A014       FI
   A024   FL    P-ORDER-NUM1                      A014       FI    SK
   A025   FL    P-ORDER-NUM2                      A014       FI    SK
   A026   FL                                      A014       FI
   ACT    FL    ACTIVITY                          PKEY       GR
   ACTV   FI    LABOR-MATERIAL-ACTIVITY           A027       PR
   ADDR   GR    CUSTOMER-ADDRESS                  CUST       FI
   BUYR   FL    BUYER                             INVO       FI
   CITY   FL                                      CUST       FI
                                                  ADDR       GR
   CSTN   FL    CUSTOMER-NUMBER                   CUST       FI    PK
   CURH   FL    CURRENT-HOURS                     ACTV       FI
   CURM   FL    CURRENT-MATERIAL                  ACTV       FI
   CUST   FI    CUSTOMER-FILE
   DATE   GR    DATE-OF-PURCHASE                  INVO       FI
   DAY    FL                                      INVO       FI
                                                  DATE       GR
   DESC   FL    ITEM-DESCRIPTION                  INVT       FI
   ESTH   FL    EST-HOURS                         ACTV       FI
   ESTM   FL    EST-MATERIAL                      ACTV       FI
```

Figure 5-7. DD-990 Cross-Reference Report (Entire Dictionary File) (Sheet 1 of 3)

DD-990 CROSS REFERENCE REPORT L. V. E. YYDDD

ID	TYPE	NAME	USED IN	TYPE	KEY
INUM	FL	INVOICE-NUMBER	INVO	FI	PK
INVO	FI	INVOICE-FILE	A028	PR	
INVT	FI	INVENTORY-FILE	A027	PR	
JOB	FL	JOB-NUMBER	PKEY	GR	
MNTH	FL	MONTH	INVO	FI	
			DATE	GR	
NAME	FL	CUSTOMER-NAME	CUST	FI	
PART	FL	PART-NUMBER-ORDERED	INVO	FI	SK
PERH	FL	PERCENT-LABOR-COMPLETED	ACTV	FI	
PERM	FL	PERCENT-MATERIAL	ACTV	FI	
PKEY	FL	ACTIVITY-PRIMARY-KEY	ACTV	FI	PK
PRIC	FL	UNIT-PRICE	INVT	FI	
PTNO	FL	PART-NUMBER	INVT	FI	PK
QUAN	FL	QUANTITY-ORDERED	INVO	FI	
QUOH	FL	QUANTITY-ON-HAND	INVT	FI	
STAT	FL		CUST	FI	
			ADDR	GR	
STRE	FL	STREET	CUST	FI	
			ADDR	GR	
TAG1	FL		INVO	FI	
TAG2	FL		INVO	FI	
TODH	FL	TODATE-HOURS	ACTV	FI	
TODM	FL	TODATE-MATERIAL	ACTV	FI	
YEAR	FL		INVO	FI	
			DATE	GR	
ZIP	FL		CUST	FI	

Figure 5-7. DD-990 Cross-Reference Report (Entire Dictionary File) (Sheet 2 of 3)

```
DD-990 CROSS REFERENCE REPORT                    L. V. E.  YYDDD
   NAME                                 ID
   ---------------------------------    ----
   ACTIVITY                             ACT
   ACTIVITY-PRIMARY-KEY                 PKEY
   BUYER                                BUYR
   CARD-FILE                            A001
   CD-CUSTOMER-NBR                      A011
   CD-PUNCH                             A009
   CD-SALES-AMT                         A012
   CD-SALESMAN-NBR                      A010
   CURRENT-HOURS                        CURH
   CURRENT-MATERIAL                     CURM
   CUSTOMER-ADDRESS                     ADDR
   CUSTOMER-FILE                        CUST
   CUSTOMER-NAME                        NAME
   CUSTOMER-NUMBER                      CSTN
   DATE-IN                              A004
   DATE-OF-PURCHASE                     DATE
   DAY-IN                               A006
   EST-HOURS                            ESTH
   EST-MATERIAL                         ESTM
   INVENTORY-FILE                       INVT
   INVOICE-FILE                         INVO
   INVOICE-NUMBER                       INUM
   ITEM-DESCRIPTION                     DESC
   JOB-NUMBER                           JOB
   LABOR-MATERIAL-ACTIVITY              ACTV
   MONTH                                MNTH
   MONTH-IN                             A005
   P-AMOUNT-PAID1                       A020
   P-AMOUNT-PAID2                       A022
   P-CONT-AMT                           A019
   P-CUST-NUMBER                        A015
   P-EXP-DATE                           A017
   P-EXPIRES-DATE                       A016
   P-ORDER-NUM1                         A024
   P-ORDER-NUM2                         A025
   P-TERMS-AGREED                       A018
   P-TYPE-PAYMENT1                      A021
   P-TYPE-PAYMENT2                      A023
   PART-NUMBER                          PTNO
   PART-NUMBER-ORDERED                  PART
   PAYMENTS-RECEIVED-FILE               A014
   PERCENT-LABOR-COMPLETED              PERH
   PERCENT-MATERIAL                     PERM
   PUNCH                                A003
   QUANTITY-ON-HAND                     QUOH
   QUANTITY-ORDERED                     QUAN
   SEQ-RR-FILES-KEY                     A000
   STREET                               STRE
   TODATE-HOURS                         TODH
   TODATE-MATERIAL                      TODM
   UNIT-PRICE                           PRIC
   YEAR-IN                              A007
     TOTAL NUMBER OF ENTITIES REPORTED =     67
```

Figure 5-7. DD-990 Cross-Reference Report (Entire Dictionary File) (Sheet 3 of 3)

5.5 LIST FILE — LSTFIL

The LSTFIL utility generates file descriptions from the dictionary file in the proper DDL syntax. The following screen appears for LSTFIL:

```
DATA DICTIONARY LIST FILE <VERSION L.V.E. YYDDD>

                    PASSWORD:
                FILE NAME/ID:
          LISTING ACCESS NAME:
                  MODE(F,B):
```

Respond to the prompts as follows:

PASSWORD
: This prompt appears only if DBMS-990 security is installed on the system. Enter the previously assigned master password.

FILE NAME/ID
: Enter the name or ID of the file to be listed. If you do not respond to this prompt, the listing access name must contain a pathname for a directory. In this case, the LSTFIL utility will output all of the file definitions in the dictionary. A sequential file is created in the directory specified for each file definition in the dictionary. The directory specified must be large enough to contain an entry for each file definition in the dictionary file.

LISTING ACCESS NAME
: Enter the name of a device where the report will be output.

MODE(F,B)
: Enter B to execute LSTFIL in background mode. Enter F to execute LSTFIL in foreground mode.

The LSTFIL utility generates a listing of all file definitions in the dictionary file. The following screen generates an example of a file definition generated by LSTFIL. This was originally generated with the Automatic File Definition utility as shown in Figure 5-8.

```
LIST DDL FOR A FILE  <L.V.E.  YYDDD>

                    PASSWORD:  <master password>
                FILE NAME/ID:  PAYMENTS-RECEIVED-FILE
          LISTING ACCESS NAME:  <pathname of output file>
                  MODE (F,B):  F
```

```
FILE=A014, TYPE=KIF
NAME=PAYMENTS-RECEIVED-FILE
*
ID=A015=CN/8. 0, DUP=N, MOD=N
NAME=P-CUST-NUMBER
*
LINE=01
NAME=PAYMENTS-RECORD
FIELD=A015=CN/8. 0
NAME=P-CUST-NUMBER
GROUP=A016
NAME=P-EXPIRES-DATE
FIELD=A017=IS/2
NAME=P-EXP-DATE
ENDG
FIELD=A018=CS/3. 0
NAME=P-TERMS-AGREED
FIELD=A019=CN/8. 2
NAME=P-CONT-AMT
FIELD=A020=PK/8. 2
NAME=P-AMOUNT-PAID1
FIELD=A021=CH/1
NAME=P-TYPE-PAYMENT1
FIELD=A022=PK/8. 2
NAME=P-AMOUNT-PAID2
FIELD=A023=CH/1
NAME=P-TYPE-PAYMENT2
FIELD=A024=CH/10
NAME=P-ORDER-NUM1
FIELD=A025=CH/10
NAME=P-ORDER-NUM2
FIELD=A026=CH/5
ENDL
*
SECONDARY-REFERENCES
A016, DUP=Y, MOD=Y
A024, DUP=Y, MOD=Y
A025, DUP=Y, MOD=Y
END.
```

Figure 5-8. LSTFIL Output

5.6 GENERATE COPY BOOK — GCB

The GCB utility generates a COBOL description for a file defined in the dictionary file. The output from GCB can be included in a COBOL program following an FD statement. The following screen appears for GCB:

```
GENERATE COPYBOOK <VERSION L.V.E. YYDDD>

                        PASSWORD:
                    FILE NAME/ID:
            LISTING ACCESS NAME:
                    MODE(F,B):
```

Respond to the prompts as follows:

PASSWORD
: This prompt appears only if DBMS-990 with security is installed on the system. Enter the assigned master password.

FILE NAME/ID
: Enter the name or ID of the file in the dictionary that the GCB procedure will use. If you do not respond to this prompt, a COBOL file definition is generated for each file defined in the dictionary.

LISTING ACCESS NAME
: Enter the name of a device where the report will be output. If you did not respond to the prompt FILE NAME/ID, the listing access name must contain a pathname for a directory. In this case, the GCB procedure outputs all of the file definitions in the dictionary. A sequential file will be created in the directory specified for each file definition in the dictionary. The directory specified must be large enough to contain an entry for each file definition in the dictionary file.

MODE(F,B)
: Enter B to execute GCB in background mode. Enter F to execute GCB in foreground mode.

An additional feature of GCB is the filler option. This option can generate COBOL data definitions that include fillers. The field must be given a long name using IDL or BDL with the following syntax:

```
FILLER-<COBOL data type>-<length>
```

If the following fields were included in a file definition

```
FIELD = B022 = CH/5 ,NAME = NAME
FIELD = B023 = CH/80 ,NAME = FILLER-X-80
FIELD = B024 = CH/101,NAME = ADDR
```

GCB output would be as follows:

```
03  NAME      PIC X(5)
03  FILLER    PIC X(80)
03  ADDR      PIC X(101)
```

The GCB utility generates COBOL record/data descriptions from the definitions in the dictionary file. The following screen generates an example of a sequential file output by the GCB utility as shown in Figure 5-9. This particular example is the same file used in Figure 4-4 in the DDL explanation.

```
GENERATE COPYBOOK  <L.V.E.   YYDDD>

            PASSWORD: <master password>
       FILE NAME/ID: ACTV
LISTING ACCESS NAME: <pathname of output file>
         MODE(F,B): F


*
* FILE: ACTV
* GENERATED BY DD-990              07/31/81    13:23:55
*
 01 ACTV-AA.
     03 ACTIVITY-PRIMARY-KEY.
        05 JOB-NUMBER                          PIC 9(5).
        05 ACTIVITY                            PIC 9(3).
     03 EST-HOURS                              PIC S9(6)V9 COMP-3.
     03 CURRENT-HOURS                          PIC S9(6)V9 COMP-3.
     03 TODATE-HOURS                           PIC S9(9)V9(2) COMP-3.
     03 PERCENT-LABOR-COMPLETED                PIC 9(3)V9.
     03 EST-MATERIAL                           PIC S9(9)V9(2) COMP-3.
     03 CURRENT-MATERIAL                       PIC S9(9)V9(2) COMP-3.
     03 TODATE-MATERIAL                        PIC S9(9)V9(2) COMP-3.
     03 PERCENT-MATERIAL                       PIC 9(3)V9.
```

Figure 5-9. GCB Output

6

Data Manager and Query-990

6.1 INTRODUCTION

The combination of the DD-990 system and Query-990 allows total control and access of a data processing environment. This section discusses the procedures and file structures necessary to allow DD-990 to access conventional files using Query-990. This is possible through the data manager, a component of DD-990, which interfaces between Query-990 and conventional files. The data manager allows Query-990 to access key indexed, relative record, and sequential files. The data manager is not necessary for access to data base files. DBMS-990 is required for Query-990 access to data base files. Query-990 is a nonprocedural language using logical statements to generate reports and access files. More information on Query-990 can be found in the *Query-990 User's Guide*. Query-990 examples using conventional files are included at the end of this section.

6.2 DATA MANAGER PROCEDURES

The data manager provides the interface to Query-990 for DD-990. There are several data manager procedures to be used with Query-990. They are as follows:

- Start Data Manager (SDM) — Starts the data manager task.
- Assign File ID (AFID) — Associates a pathname with a conventional file defined in the dictionary. A file must be assigned in order to be accessed by Query-990.
- List Assigned Files (LAF) — Produces a listing of conventional files currently assigned to the data manager.
- Release File ID (RFID) — Removes the association between a pathname and a file defined in the dictionary.
- End Data Manager (EDM) — Terminates the data manager task.

6.2.1 Start Data Manager — SDM

Enter the SDM command to bid the data manager task to accept Query-990 calls. No prompts appear. If DBMS is installed on the system, DBMS must be running and the dictionary file must be assigned before executing the SDM command.

NOTE

The logical name S$QUERY must be previously assigned under a DNOS system.

6.2.2 Assign File ID — AFID

Enter the AFID command to associate a pathname with a file name or ID defined in the dictionary. The following screen appears:

```
ASSIGN FILE ID <VERSION: L.V.E. YYDDD>

          FILE NAME/ID:
       FILE PATHNAME:
          FILE ACCESS:
```

Respond to the prompts as follows:

FILE NAME/ID
: Enter the file name (1 to 30 characters long) or the file ID (1 to 4 characters long) to identify the file in the dictionary.

FILE PATHNAME
: Enter the operating system pathname that contains the data. If you do not specify a pathname, the dictionary uses the pathname assigned to the file definition through IDL. This pathname must be assigned to a previously created file.

FILE ACCESS
: The file access specifies how the data manager will open the file. The only access methods allowed are EXCL (exclusive) and SHRD (shared). If you specify EXCL, the data manager opens the file for exclusive write accesses. The data manager retains this access to the file until you enter the Release File ID (RFID) command for the file. As long as the file is assigned to the data manager, other applications can access the file for read-only access but not for update.

If you need to run applications that update a file while that file is assigned to the data manager, specify SHRD access when the file is assigned.

The EXCL access results in the most efficient use of the data manager. Access to relative record and sequential files through the data manager is slower when you use SHRD access.

If you assign EXCL access to the file, queries using either SHRD or EXCL have access to the file. However, if you assign SHRD access to the file, only queries using SHRD access can be executed. Access is an optional clause in Query-990. The default is EXCL.

You can execute update and delete queries against KIFs and relative record files whether the access specified in the query is EXCL or SHRD. In contrast, you can execute update and delete queries against sequential files only when the query itself specifies EXCL access. Therefore, sequential files must be assigned EXCL access if the file is to be updated.

6.2.3 List Assigned Files — LAF

The LAF command produces a listing of files assigned to the data manager. When you enter the LAF command, the following screen appears:

```
<VERSION: L.V.E. YYDDD>
LIST ASSIGNED FILES

          MASTER PASSWORD:
LISTING ACCESS NAME:
          MODE(F, B):
```

Respond to the prompts as follows:

MASTER PASSWORD
: Enter the four-character alphanumeric value to be used as the master password of the installed system. This prompt appears only if data manager security is included in DD-990 generation.

LISTING ACCESS NAME
: Enter the name of a sequential file or device where a listing of the assigned file is output.

MODE(F, B)
: Enter B to execute LAF in background mode. Enter F to execute LAF in foreground mode.

6.2.4 Release File ID — RFID

The RFID command releases the file defined in the data dictionary from the operating system pathname. The following screen appears:

```
RELEASE FILE ID <VERSION L.V.E. YYDDD>

          FILE ID/NAME:
```

Respond to the prompt as follows:

FILE ID/NAME
: Enter the file ID (1 to 4 characters long) or the file name (up to 30 characters long) that identifies the file to the data dictionary.

6.2.5 End Data Manager — EDM

Enter the EDM command to terminate the data manager task. The following prompt appears:

```
END DATA MANAGER <VERSIONS: L.V.E. YYDDD>
          ARE YOU SURE?(Y/N):
```

Respond to the prompt as follows:

ARE YOU SURE?(Y/N)
: Enter Y if the data manager task is to be terminated. Enter N if the data manager task is to continue.

6.3 CONVENTIONAL FILE STRUCTURES

DD-990 allows Query-990 to access KIFs, relative record files, and sequential files. Almost all Query-990 syntax is allowed for conventional files. The exceptions are described in the following paragraphs. You need not specify in the Query statement which file type is being accessed.

6.3.1 Key Indexed Files (KIFs)

The structure of a KIF is almost identical to that of a data base file. A record in a KIF is all lines with primary keys having the same value. If duplicates are not allowed for primary keys, a KIF is a collection of records containing one line each. A key is anything defined to the operating system as a key when the KIF is created. Therefore, KIFs can have both primary and secondary keys. No POSITION clause is allowed for KIFs with Query-990 since duplicate occurrences of a primary key are ordered chronologically.

When defining KIFs to DD-990, you can specify a key defined earlier as a secondary key to the operating system as the primary key in the dictionary definition. In this case, the key defined as a primary key to the operating system must be defined as a secondary key in the dictionary definition. This can be used when a secondary KIF key is used as the logical primary key because COBOL does not allow duplicates on the primary key, and duplicates might be needed. The only requirement when defining a KIF to the dictionary is that all keys defined to the operating system also be defined in the dictionary.

6.3.2 Relative Record Files

In a relative record file, each record contains one line. The primary key for each record is the record number that the operating system associates with that record. Relative record files do not have secondary keys. The only restriction when using Query-990 with relative record files occurs in the POSITION clause. The POSITION clause in the INSERT function is not allowed, since only one line can exist per record.

NOTE

When a relative file organization is used, the first byte of the record cannot contain hexadecimal FF. This position is used as a delete indicator; any record containing a hexadecimal FF in the first byte will not be returned to a COBOL program or Query-990 when the file is read.

6.3.3 Sequential Files

Sequential files are treated much like relative record files. Each record contains only one line. The primary key is the record number. Sequential files do not have secondary keys.

The only restrictions when using Query-990 with sequential files occur in the POSITION clause and the DELETE function. Since operating systems do not provide any delete capability for sequential files, the DELETE and DELETE RECORD statements in Query-990 are not allowed. The POSITION clause is not allowed for sequential files with Query-990 since duplicate occurrences of a key cannot occur. Each record has only one line.

6.4 QUERY-990 EXAMPLES

Almost all Query-990 functions available for DBMS files are also available for conventional files. Query-990 is particularly useful for data access and custom reporting. The remainder of this section contains examples of Query-990 statements and the resulting output. The three file types in these examples are key indexed, relative record, and sequential. Appendix C lists these test files. For more information on writing and using Query-990, refer to the *Query-990 User's Guide.*

The following Query-990 statement lists the contents of the relative record inventory file INVT. The information for all parts in the inventory file are listed as they are organized in the file.

Query statement:

```
LIST PTNO 01 HEADER DEFAULT SKIP FROM INVT BY LIST
```

Query output:

```
LIST PTNO 01  HEADER DEFAULT CKIP FROM INVT BY LIST


PTNO      DESC              PRIC        QUOH

     1 HAMMER               5.00        2000
     2 PLIERS              10.00        1500
     3 SCREW DRIVER         3.00       12000
     4 LEVEL               50.00         100
     5 WRENCH              20.00         500
     6 MICROMETER          50.00         200
     7 SAW                 12.00         400
```

The following Query-990 statement lists the contents of the sequential file CUST. The customer line of information is listed from the file without special ordering. The format is an automatic function of Query-990.

Query statement:

```
LIST CUST-LINE FROM CUSTOMER-FILE BY LIST
```

Query output:

```
ABC HARDWARE STORE     8004 N. STREET          AUSTIN          TX  78750
HANDY ANDY HARDWARE    4000 BURNET RD.         AUSTIN          TX  78753
CAPITOL HARDWARE CO.   8153 RESEARCH BLVD.     AUSTIN          TX  78752
NORTHWEST HARDWARE     2000 LAMAR BLVD.        AUSTIN          TX  78751
```

Query-990 has the capability of manipulating information from more than one file at a time. For example, the following Query statement generates invoices from the three files, INVT, INVO, and CUSTOMER-FILE.

Query statement:

```
DEFINE COST : CN/6.2  = QUAN * PRIC;
       INVOICE-COST : CN/8.2 = RECORD TOTAL COST;

LIST 24X,"INVOICE #:",INUM
HEADER "*****************************************" SKIP 2;

5X, "DATE:", MNTH, "/", DAY, "/", YEAR HEADER SKIP 1;

"SOLD TO:", NAME HEADER SKIP;

8X, STRE; 8X, CITY, ", ", STAT, ZIP;

PART, QUAN, DESC, COST
HEADER SKIP 2  "        PART  QTY  DESCRIPTION      PRICE" SKIP;

21X "TOTAL AMOUNT" 3X INVOICE-COST HEADER SKIP FOOTING SKIP;

FROM INVO CUSTOMER-FILE INVT LINKED BY BUYR = CUST-NUMBER, PART = PTNO

BY KEY BY LIST WHERE ANY MNTH = 5
```

Query output:

```
**********************************************

                          INVOICE #:    163023

    DATE: MM/DD/YY

SOLD TO: CAPITOL HARDWARE CO.
         8153 RESEARCH BLVD.
         AUSTIN          , TX 78752

         PART  QTY   DESCRIPTION        PRICE

          5    100   WRENCH             2000.00
          2    300   PLIERS             3000.00
          6     40   MICROMETER         2000.00

                     TOTAL AMOUNT       7000.00

**********************************************

                          INVOICE #:    175571

    DATE: MM/DD/YY

SOLD TO: NORTHWEST HARDWARE
         2000 LAMAR BLVD.
         AUSTIN          , TX  78751

         PART  QTY   DESCRIPTION        PRICE

          1    400   HAMMER             2000.00
          7    100   SAW                1200.00

                     TOTAL AMOUNT       3200.00
```

Query-990 can use one file to determine the importance of information contained in other files. In the following Query-990 example, all the actual data comes from the second file, CUSTOMER-FILE. The following Query-990 statement generates a mailing list from the CUSTOMER-FILE file based on recent purchases from the INVO file:

Query statement:

```
LIST NAME; STRE; CITY, STAT, ZIP FOOTING SKIP 5;
     FROM INVO CUSTOMER-FILE LINKED BY BUYR = CUST-NUMBER
     NO HEADER BY KEY BY LIST
     WHERE ANY MNTH = 5
```

Query output:

```
CAPITOL HARDWARE CO.
8153 RESEARCH BLVD.
AUSTIN              TX   78752




NORTHWEST HARDWARE
2000 LAMAR BLVD.
AUSTIN              TX   78751
```

The following Query-990 statement totals inventory in the INVO file. This example uses the TOTAL feature in Query.

Query statement:

```
DEFINE VALUE : CN/8.2 = QUOH * PRIC;
       TOTAL-VALUE : CN/8.2 = TOTAL VALUE;

LIST DESC, PRIC, 4X, QUOH, 5X, VALUE;
     10X "TOTAL VALUE OF INVENTORY     $", 0X, TOTAL-VALUE HEADER SKIP

FROM INVO BY KEY BY LIST
HEADER SKIP "ITEM                PRICE      QTY       VALUE" SKIP
```

Query output:

ITEM	PRICE	QTY	VALUE
HAMMER	5.00	2000	10000.00
PLIERS	10.00	1500	15000.00
SCREW DRIVER	3.00	12000	36000.00
LEVEL	50.00	100	5000.00
WRENCH	20.00	500	10000.00
MICROMETER	50.00	200	10000.00
SAW	12.00	400	4800.00
TOTAL VALUE OF INVENTORY			$ 90800.00

For more information on writing and using queries, refer to the *Query-990 User's Guide*.

7

Security

7.1 INTRODUCTION

Password security is an optional feature that can be enabled during DD-990 generation. The DD-990 password security option is chosen at DDGEN time (see Section 2). This option allows passwords to be specified to protect against unauthorized Query-990 access to data contained in the files that have been defined in the dictionary file. The security is enforced only through Query-990. In deciding whether to install security in DD-990, keep in mind that security increases access time for Query-990 requests and memory requirements for the data manager.

7.2 PASSWORDS

When security is installed in DD-990, every Query must contain a valid password. However, these passwords are different from those required previously by prompts such as DDR, GCB, or the IDL. The password is the master password required by DBMS-990 security for access to the dictionary file. The passwords required by Query-990 are those passwords which have been assigned access to the conventional being querried.

The data base administrator (DBA) is responsible for controlling passwords. The DBA assigns user passwords and authorizes and determines the number of passwords needed.

7.3 ACCESS AUTHORIZATION

The DBA can limit user access to the file, line, group, or field level, or to a combination of these levels. For example, if you require access to data in the personnel file, you might be denied access to the salary field. Also, the DBA can restrict the functions (such as read, write, and delete) available to you. For example, you might be allowed to read a customer's identification number and address without being allowed to change that data.

The Add Password (ADDPSW) command assigns each user password to a set of files. Any file that is not in this set is inaccessible when this password is in use. Thus, accessing a file requires either a password that is assigned authorization to the file or the master password.

Each file in the set is also assigned an access authorization that restricts the type of function that can be performed on that file. The following access types are available:

- Read
- Write (Update)
- Add (Insert)
- Delete

These access types are combined to form your access authorization for a particular file. For example, one password may specify an authorization with only read access to a particular file, while another password specifies an authorization with read, write, and add access to that file. If you are assigned the first password, you have only read access to that file.

The DBA can use almost any combination of these access types to form a permissible access authorization. For a no access authorization, the DBA does not specify any of the access types. For example, if a file allows all authorizations, the DBA can assign no access to specific lines in the file by giving them no authorizations when executing the Add Password Entry (ADDPE) command.

The DBA can assign an authorization code to all levels of data. In the absence of an assigned authorization code, data elements assume the authorization of the next highest element. For example, lines assume the authorization code of the file. If necessary, the DBA can avoid this by assigning to a line only a subset of its file's authorization, including no access. If a file has read and write authorization, a line in that file might be assigned only read authorization.

Similarly, all fields in a line have the same authorization as the line. To avoid this, the DBA can assign to a field only a subset of the line's authorization, including no access.

To facilitate security checking, groups are treated as fields. The DBA must resolve any conflicts in authorization access between a field and its group.

NOTE

Assigning authorization to lines and fields requires more time and space for security checking. Because of this additional system overhead, the DBA should assign authorization at the file level only, whenever possible. Consequently, you should thoroughly justify requests for authorizations assigned to a line, group, or field.

To make a line, group, or field more secure than a file, the DBA can assign a subset of the file's authorization to the line, group, or field. The following priorities apply when assigning authorization:

Data Level	Priority
Field/Group	Highest
Line	
File	Lowest

The following restrictions apply to authorization assignments:

- Any authorization that includes delete or write access must also have read access.
- All lower-level authorizations must be a subset of their associated higher-level authorizations. A subset can be an identical authorization. For example, if an authorization is made at the field level, its associated line(s) must be assigned an authorization that contains the field's authorization. If the line needs no special restrictions, the line should be assigned the same authorization as the associated file.
- If the DBA assigns delete authorization at the line level, all fields and groups in the line must also have delete authorization. This restriction is necessary because a delete is performed on a line rather than on a field.

7.4 PASSWORD PROCEDURES

To add and change password information, you must enter a master password. The master password is necessary for computer operations in which one person or group controls all password authorizations.

All passwords are stored in security files of the data manager in a coded format. When installed with DBMS-990, changes to the password information can occur only when DBMS-990 is running. When installed without DBMS-990, changes to the password information can occur only when the dictionary file has been assigned. Security files are stored on the system disk under the directory .DBLIB on DX10 operating systems. On DNOS operating systems, the security files are installed under the directory S$DBMS.

The system checks security whenever you execute a Query on a file or files defined in the dictionary file.

CAUTION

Any change in the password entries immediately affects the DD-990 security checking. Therefore, use caution when changing password entries while Query-990 is running. If an error occurs during password assignment, an error message occurs.

7.4.1 Change Master Password (CMPSW)

To change the master password, enter the SCI command Change Master Password (CMPSW). The following prompts appear:

```
CHANGE MASTER PASSWORD
      OLD MASTER PASSWORD:
      NEW MASTER PASSWORD:
```

Respond to the prompts as follows:

OLD MASTER PASSWORD
Enter the current master password.

NEW MASTER PASSWORD
Enter four alphanumeric characters that will become the new master password.

7.4.2 Change Password (CPSW)

To change a password, enter the SCI command Change Password (CPSW). The following prompts appear:

```
CHANGE PASSWORD
            MASTER PASSWORD:
               OLD PASSWORD:
               NEW PASSWORD:
```

Respond to the prompts as follows:

MASTER PASSWORD
Enter the current master password.

OLD PASSWORD
Enter the password to be changed.

NEW PASSWORD
Enter the four alphanumeric characters that will replace the old password.

Although this procedure may take a considerable amount of time, it is more efficient than recreating all of the password information.

7.4.3 Add Password (ADDPSW)

To add a password, enter the SCI command Add Password (ADDPSW). The following prompts appear:

```
ADD PASSWORD
            MASTER PASSWORD:
                   PASSWORD:
```

Respond to the prompts as follows:

MASTER PASSWORD
Enter the current master password.

PASSWORD
Enter four alphanumeric characters that will be assigned as a new user password.

This command enters the specified password into the security file. Then, use the Add Password Entry (ADDPE) command to enter specific authorizations for the password.

7.4.4 Add Password Entry (ADDPE)

To add an entry to a password, use the SCI command Add Password Entry (ADDPE). This command includes two prompting screens. You can repeat these screens to make any number of entries to the password in a given ADDPE command. The screens appear as follows:

```
ADD PASSWORD ENTRY
                    MASTER PASSWORD:
                           PASSWORD:
          TYPE (FILE, LINE, ITEM):
                               FILE:
                               LINE:
        ITEM (FIELD OR GROUP):

 AUTHORIZATION
                       READ ACCESS?:  NO
                      WRITE ACCESS?:  NO
                        ADD ACCESS?:  NO
                     DELETE ACCESS?:  NO
                      MORE ENTRIES?:  NO
```

Respond to the prompts as follows:

MASTER PASSWORD
Enter the current master password.

PASSWORD
Enter a previously assigned user password.

TYPE (FILE, LINE, ITEM)
Enter the type of addition: F for file, L for line, or I for item (which refers to a field or group).

FILE
If the response to the type prompt is F (file), enter the file ID to be added. Otherwise, enter the file ID of the entry to be added. (A four-character value is expected.)

LINE
If the response to the type prompt is F (file), this entry is ignored (enter a carriage return). If the type is L (line), enter the line to be added. If the type is I (item), enter the line number of the entry to be added. (A two-character value is expected.)

ITEM (FIELD OR GROUP)
If the response to the type prompt is F (file) or L (line), this entry is ignored (enter a carriage return). If the type is I (item), enter the field or group ID to be added. (A four-character value is expected.)

READ ACCESS?
A YES response assigns read authorization to the entry. A NO response prohibits read access to the entry when using this password.

WRITE ACCESS?
A YES response assigns write authorization to the entry. A NO response prohibits write access to the entry when using this password. Note that if you enter a YES response, read access must also be assigned.

ADD ACCESS?
A YES response assigns add authorization to the entry. A NO response prohibits add access to the entry when using this password.

DELETE ACCESS?
A YES response assigns delete authorization to the entry. A NO response prohibits delete access to the entry when using this password. Note that if you enter a YES response, read access must also be assigned.

MORE ENTRIES?
A YES response allows more entries to be added for authorization. You can repeat the ADDPE screens any number of times until you enter NO in response to the MORE ENTRIES? prompt.

Add only one element to the password for each password entry request. For example, if line 02 needs read authorization and the file containing it needs read and write authorization, the required entries are as follows:

```
ADD PASSWORD ENTRY
                MASTER PASSWORD:  XXXX
                       PASSWORD:  YYYY
      TYPE (FILE, LINE, ITEM):    FILE
                          FILE:   FILX
                          LINE:
    ITEM (FIELD OR GROUP):

AUTHORIZATION
                    READ ACCESS?: YES
                   WRITE ACCESS?: YES
                     ADD ACCESS?: NO
                  DELETE ACCESS?: NO
                   MORE ENTRIES?: YES
```

```
ADD PASSWORD ENTRY
                MASTER PASSWORD:  XXXX
                       PASSWORD:  YYYY
      TYPE (FILE, LINE, ITEM):    LINE
                          FILE:   FILX
                          LINE:   02
    ITEM (FIELD OR GROUP):

AUTHORIZATIONS
                   READ ACCESS?:  YES
                  WRITE ACCESS?:  NO
                    ADD ACCESS?:  NO
                 DELETE ACCESS?:  NO
                  MORE ENTRIES?:  NO
```

The assignment of file authorization must precede the assignment of line authorization in the file. Similarly, line authorization must precede field/group authorization. The line authorization must be a subset of the file authorization, and the field/group authorization must be a subset of the line authorization.

7.4.5 Delete Password (DELPSW)

To delete a password and all associated information, enter the SCI command Delete Password (DELPSW). The following prompts appear:

```
DELETE PASSWORD
                MASTER PASSWORD:
                       PASSWORD:
```

Respond to the prompts as follows:

MASTER PASSWORD
: Enter the current master password.

PASSWORD
: Enter the user password to be deleted.

7.4.6 Delete Password Entry (DELPE)

To delete an entry in the password, use the SCI command Delete Password Entry (DELPE). The following prompts appear:

```
DELETE PASSWORD ENTRY
                MASTER PASSWORD:
                       PASSWORD:
      TYPE (FILE, LINE, ITEM):
                          FILE:
                          LINE:
    ITEM (FIELD OR GROUP):
```

Respond to the prompts as follows:

MASTER PASSWORD
Enter the current master password.

PASSWORD
Enter the user password of the entry to be deleted.

TYPE (FILE, LINE, ITEM)
Indicate the type of entry to be deleted: F for file ID, L for line ID, or I for field or group ID.

FILE
If the response to the type prompt is F (file), enter the identifier (ID) of the file to be deleted from the password security information. Otherwise, enter the file ID of the entry to be deleted.

LINE
If the response to the type prompt is F (file), this prompt is ignored (enter a carriage return). If the type is L (line), enter the ID of the line to be deleted. If the type is I (item), enter the line ID of the entry to be deleted.

ITEM (FIELD OR GROUP)
If the response to the type prompt is F (file) or L (line), this prompt is ignored (enter a carriage return). If the type is I (item), enter the ID of the group or field to be deleted from the password security information.

You can specify one element per DELPE command. If you delete a line, all field and group entries associated with it are also deleted. If you delete a file entry, all lines associated with it are deleted.

7.4.7 Map Password File (MPSWF)

Use the SCI command Map Password File (MPSWF) to obtain information about the contents of the password file. The following prompts appear:

```
MAP PASSWORD FILE
          MASTER PASSWORD:
     LISTING ACCESS NAME:
```

Respond to the prompts as follows:

MASTER PASSWORD
Enter the current master password.

LISTING ACCESS NAME
Enter a standard pathname or device to which the output is written.

A sample output file is as follows:

```
MAP OF PASSWORD FILE          AUTHORIZATION SYMBOLS

CREATED: 02/11/78                        R    READ ACCESS
PASSWORDS ASSIGNED:  2                   W   WRITE ACCESS
PASSWORDS AVAILABLE: 23                  A     ADD ACCESS
                                         D  DELETE ACCESS
                                         N      NO ACCESS


PASSWORD     FILE     LINE     GROUP/FIELD     AUTHORIZATON

  XXXX
               FIL1                                RWAD
                        02                          R
                                    ITM1            N
                        04                          RW
  YYYY
               FIL2                                RWA
               FIL3                                RD
```

The authorization column contains a one-character value for each access authorization assigned. The values appear in the upper right-hand corner of the listing.

7.5 ERROR MESSAGES

Table 7-1 lists and explains the error messages displayed during password procedures.

Table 7-1. Procedure Error Messages

Message	Meaning
CANNOT GET ACCESS	Another task has access to the security files at this time; try later.
CANNOT OPEN FILES	The file-access checking buffer is full; try later.
CODE CONFLICT	The entry to be added is not a subset of the authorization of the higher-level entry, or the item to be added does not have delete authority but the associated line does have delete authority.
DATABASE DOWN	SDBMS has not been executed in a DBMS environment. Otherwise, DD-990 has not been assigned.
DELETE ACCESS IMPLIES READ ACCESS-TRY AGAIN	If delete authorization is requested, read access must also be assigned.

Table 7-1. Procedure Error Messages (Continued)

Message	Meaning
DUPLICATE FILE	Tried to add a file that has already been assigned.
DUPLICATE ITEM	Tried to add an item that has already been assigned.
DUPLICATE LINE	Tried to add a line that has already been assigned.
DUPLICATE PASSWORD	Tried to add a password that has already been assigned.
FILE NOT FOUND	The file specified was not previously assigned.
INVALID CODE	Error when entering the authorizations for the entry. Reenter the command.
INVALID FILE	The file specified is longer than four characters.
INVALID FUNCTION	Check to see that the procedure has not been altered.
INVALID ITEM	Item name is longer than four characters.
INVALID LINE	The line specified is longer than two characters.
INVALID MASTER	Master password is longer than four characters.
INVALID MAX VALUE	When creating security, the MAX PASSWORDS or MAX ENTRIES prompt value was greater than 999999.
INVALID PASSWORD	The password specified is longer than four characters.
INVALID TYPE	Type is not F, L, or I.
ITEM NOT FOUND	The item specified was not previously assigned.
LINE NOT FOUND	The line specified was not previously assigned.
NO BUFFERS	The interface buffers are full; try later.
PASSWORD AREA FULL	The maximum number of passwords specified in DDINS has been exceeded.
PASSWORD NOT FOUND	Tried to delete a password that was not assigned previously.
WRITE ACCESS IMPLIES READ ACCESS-TRY AGAIN	If write authorization is requested, read access must also be assigned.

7.5.1 Errors Returned from DD-990

If DD-990 returns an unexpected status code while performing a Query, the following error message occurs:

INVALID STATUS CODE: XX

where:

XX is the status returned from the Query.

7.5.2 System Log Message (Optional)

A message is printed on the system log whenever Query returns a security violation (SV status code). The format is as follows:

JJJ:HHMM DBMS SV T (task) = II (RR):SS

where:

JJJ is the Julian date.

HHMM is the time of error (24-hour clock).

II is the task-installed ID.

RR is the task run-time ID.

SS is the station ID of the terminal associated with the task.

8

Error Messages

8.1 INTRODUCTION

DD-990 errors consist of the following:

- BDL errors
- IDL errors
- Utility errors
- SDM errors
- DDSTAT errors
- AFD errors

8.2 BDL ERROR MESSAGES

The Batch Data Librarian generates the following messages:

0001 ERROR(S) IN THE DDL.

Explanation:
This is an informative message indicating the number of errors in the BDL input file syntax.

User Action:
If the number of errors is not zero, correct any errors and resubmit the input file.

0008 INVALID LENGTH WAS SPECIFIED

Explanation:
Some data types have fixed lengths. If the length is specified, it must be one of these previously set values. The following are default lengths for these data types:

IS = 2
LG = 2
ID = 4
RS = 4
RD = 8

User Action:
Correct the error in the data type length and resubmit the input file.

0010 INVALID CHARACTERS AT END OF NAME

Explanation:
The name cannot end with one of the following invalid characters: -, or __.

User Action:
Correct the name and resubmit the input file.

0011 LENGTH OF NAME EXCEEDS 30 CHARACTERS

Explanation:
The name must be from 1 to 30 characters long.

User Action:
Reduce the length of the name and resubmit the input file.

0012 DESCRIPTION LINES EXCEED 160 CHARACTERS

Explanation:
The descriptive text must be less than 160 characters in length. (Blanks at the end of the line are not counted as characters.)

User Action:
Reduce the size of the text and resubmit the input file.

0013 UNEXPECTED EOF

Explanation:
An end-of-file (EOF) was read on the file before the END. statement was read.

User Action:
Add an END. statement as the last statement in the input file.

0014 INVALID ID OR ID LONGER THAN 4 CHARACTERS

Explanation:
The ID cannot be longer than four characters or contain invalid characters.

User Action:
Check the length of the ID. Be sure that the ID contains only alphabetic and/or numeric characters.

0015 INVALID DATA TYPE

Explanation:
An unknown data type was specified.

User Action:
Check Appendix B of this manual for valid data types.

0016 INVALID DECIMAL COUNT

Explanation:
A decimal greater than the length of the field was specified. The decimal count must be less than or equal to the length.

User Action:
Correct the error and resubmit the input file.

0017 NUMERIC FIELD EXPECTED

Explanation:
Nonnumeric characters were found where a numeric field was expected.

User Action:
Correct the error and resubmit the input file.

0018 Y OR N EXPECTED

Explanation:
Only Y or N are valid characters following the equal sign.

User Action:
Correct the error and resubmit the input file.

0019 KEY LENGTH EXCEEDS 100 CHARACTERS

Explanation:
The specified key size for a key indexed file (KIF) must be from 1 to 100 characters long.

User Action:
Correct the key size of the file and resubmit the input file.

0020 TAG FIELD MUST BE DATA TYPE IS OR CH

Explanation:
Only integer (IS) or character (CH) data types are supported for tag fields.

User Action:
Change the field data type to IS or CH and resubmit the input file. The data file must contain either integer or character data at this position in the file.

0021 INVALID SYNTAX

Explanation:
BDL is unable to decode the key word. Either an invalid key word was entered, or the key word clause is incomplete.

User Action:
Correct the error and resubmit the input file.

0022 ID SHOULD BE DEFINED FIRST

Explanation:
The ID statement identifying the primary key is required before the LINE statement.

User Action:
Add the ID statement and resubmit the input file.

0023 DUPLICATE FIELD ID, SEE ?1

Explanation:
A field ID can exist only once in the same file definition. The only exception to this is the primary key in a KIF. The ID specified is included more than once in a file definition.

User Action:
Change one of the IDs and resubmit the input file.

0024 INVALID LENGTH OF LINE ID

Explanation:
The line ID must be two characters long.

User Action:
Correct the length of the line ID and resubmit the input file.

0025 DUPLICATE LINE ID

Explanation:
The line IDs must be unique within a file definition.

User Action:
Change one of the duplicate IDs and resubmit the input file.

0026 DUPLICATE GROUP ID

Explanation:
A group ID can appear only once in the same file definition.

User Action:
Change one of the IDs and resubmit the input file.

0027 UNDEFINED FIELD OR GROUP ID

Explanation:
A field or group specified as a secondary key does not exist in the file definition.

User Action:
Add the ID to the file definition and resubmit the input file.

0028 DUPLICATE FILE DEFINITION

Explanation:
An attempt was made to add a file definition that already exists in the dictionary file.

User Action:
If the current file is no longer valid, use the Interactive Data Librarian (IDL) the current file definition from the dictionary. If the current file is valid, change the file ID for the definition to be added.

0029 INVALID FILE TYPE

Explanation:
An invalid file type was specified. Valid file types include DB, KIF, RR, and SEQ.

User Action:
Enter a valid file type and resubmit the input file.

0030 NESTED GROUPS ARE NOT ALLOWED

Explanation:
A group within another group cannot be included in a file definition. Only single-level groups are allowed.

User Action:
Remove the second GROUP statement, or insert an END statement before the second GROUP statement.

0031 END. EXPECTED

Explanation:
An end-of-file (EOF) was read on the input file before the END statement was processed.

User Action:
Add an END statement as the last statement of the input file.

0032 INVALID VOL SPECIFIED FOR KEY FIELD

Explanation:
Zero was specified for the number of keys. The number of keys must be at least one.

User Action:
Enter the correct number of keys and resubmit the input file.

0033 VALUE CLAUSE IS REQUIRED IN LINE DEFINITION

Explanation:
If TAG was specified with the FILE statement, a VALUE clause must be specified in each line definition.

User Action:
Select the field in each line that is the tag field and add a VALUE clause. The tag field must be at the same offset in each line and must be the same data type and length in each line.

0034 DATA TYPE NOT SPECIFIED

Explanation:
The data type for the field ID must be specified with the FIELD statement.

User Action:
Select one of the data types from Appendix B in this manual.

0035 ZERO LENGTH SPECIFIED

Explanation:
Zero is an invalid length for all data types.

User Action:
Select a data type length greater than zero and resubmit the input file.

0039 PRIMARY KEY MUST BE INCLUDED IN LINE ?1

Explanation:
In a KIF, the primary key must be included in the same position in each line.

User Action:
Add the primary key to the line indicated and resubmit the input file.

0040 PRIMARY KEY OFFSETS DO NOT MATCH

Explanation:
The primary key for a KIF must be at the same offset in each line.

User Action:
Correct the error and resubmit the input file.

0041 TAG FIELD OFFSETS DO NOT MATCH

Explanation:
Tag fields must be at the same offset in each line.

User Action:
Correct the error and resubmit the input file.

0042 MODIFIABLE AND NON MODIFIABLE KEYS OVERLAP, SEE ?1

Explanation:
Secondary keys that overlap each other must have the same modifiable characteristic; that, is if one is modifiable, the other must also be modifiable. When a modifiable key value is updated, the key portion is replaced. Therefore, if one key is part of another key, both keys must have the option to be updated.

User Action:
Correct the problem and resubmit the input file.

0043 PRIMARY KEY CANNOT BE A SECONDARY KEY

Explanation:
A primary key cannot also be defined as a secondary key.

User Action:
Remove the primary key ID from the list of secondary keys and resubmit the input file.

0044 LINE LENGTH EXCEEDS MAXIMUM ALLOWABLE LENGTH, SEE ?1

Explanation:
The maximum line length for a data base file is 512 characters.

User Action:
Ensure that the total number of bytes in the line is less than 512 bytes. The total number of bytes equals ten plus the length of the primary key plus the length of the fields plus eight bytes per secondary key in the line.

0045 MORE THAN 200 FIELD/GROUPS DEFINED

Explanation:
The maximum number of fields and groups per file definition for a data base file is 200 minus the number of line types in the file minus the number of keys in the file.

User Action:
Reduce the number of fields or groups, the number of line types, or the number of keys in the file definition.

0046 LINE= MUST BE SPECIFIED

Explanation:
The LINE= clause specifying the number of data lines must be specified for a data base file.

User Action:
Correct the error and resubmit the input file.

0047 PRIMARY KEY CANNOT BE DEFINED IN THE LINE

Explanation:
The primary key cannot be included in the line definition for a file type other than KIF.

User Action:
Remove the primary key from the line definition and resubmit the input file.

0049 INVALID ACCESS MODE SPECIFIED

Explanation:
The access mode for the key is incorrect. Only RANDOM/1 and SEQUENTIAL/1 are allowed.

User Action:
Correct the error and resubmit the input file.

0050 FIELD ATTRIBUTE DOES NOT MATCH DEFINED ATTRIBUTE

Explanation:
The field already exists in the dictionary and the attribute specified does not match the value stored in the dictionary.

User Action:
Correct the error and resubmit the input file.

0051 GROUP ATTRIBUTES DO NOT MATCH DEFINED ATTRIBUTES

Explanation:
The group ID already exists in the dictionary file, and the group currently being processed does not match the one in the dictionary. (One function of BDL is to verify that the fields in the group being processed are the same as those already defined and that they occur in the same order.)

User Action:
Verify that the group in the input file is the same as the one in the dictionary or change the name of the group ID to one that does not already exist.

0052 FILE ALREADY DEFINED

Explanation:
An attempt was made to add a file definition for a file ID that already exists.

User Action:
Either change the file ID or delete the file definition from the dictionary.

0053 FILE WAS NOT ADDED TO THE DATA DICTIONARY

Explanation:
This error occurs at the end of the BDL listing when any errors were detected and the file is not added to the dictionary file.

User Action:
Correct any errors and resubmit the input file.

0054 INVALID TAG FIELD DATA FORMAT

Explanation:
The data type of this tag field does not match the data type of the tag field for the previous line. Tag fields must be of the same data type.

User Action:
Correct the error and resubmit the input file.

0055 INVALID TAG FIELD LENGTH

Explanation:
The length of this tag field does not match the length of the tag field for the previous line. Tag fields must have the same length.

User Action:
Correct the error and resubmit the input file.

0058 SECONDARY REFERENCE IS NOT ALLOWED FOR RR OR SEQUENTIAL FILE TYPES

Explanation:
Secondary keys were specified for a relative record or sequential file. Only data base files and KIFs can contain secondary keys.

User Action:
Delete the SECONDARY_REFERENCES line and the list of secondary keys following it. Resubmit the input file.

0059 PRIMARY KEY MUST BE DATA TYPE ID FOR RR OR SEQUENTIAL FILE TYPES

Explanation:
The primary key for a relative record or sequential file must be a record number that is by definition a double precision integer (data type ID).

User Action:
Make the data type for the primary key data type ID or use the default field A000.

0060 MORE THAN 13 SECONDARY KEYS WERE DEFINED

Explanation:
The maximum number of secondary keys allowed is 13.

User Action:
Delete the extra secondary keys and resubmit the input file.

0061 DB FILE MAY NOT CONTAIN TAGS

Explanation:
Tag fields are not supported for data base files.

User Action:
Correct the error and resubmit the input file.

0062 VALUE CLAUSE IS NOT ALLOWED UNLESS 'TAG' IS SPECIFIED IN FILE STATEMENT

Explanation:
The FILE statement must contain the TAG clause if tags are included in the file.

User Action:
Either add the TAG clause to the FILE statement or remove the VALUE clause from the FIELD statement.

0063 INVALID CHARACTERS IN NAME

Explanation:
Valid characters are the alphabetics, numerics, the dollar sign $, the underscore _, and the dash -.

User Action:
Include only valid characters in the name and resubmit the input file.

0064 FIELD LENGTH MUST BE SPECIFIED

Explanation:
Only the data types IS, ID, RS, RD, and LG do not require the field length to be specified because they have a default length.

User Action:
Use the following format to add a length to the FIELD statement.

'FIELD = XXXX = <data type>/<length>.<decimal count>'

Also, see Appendix B in this manual.

0065 INVALID KEYWORD FOR FILE TYPE DB

Explanation:
The key words used to describe a key for data base file are ACCESS and VOL.

User Action:
Correct the error and resubmit the input file.

0066 INVALID KEYWORD FOR FILE TYPE KIF

Explanation:
The key words used to describe a key for a KIF are DUP and MOD.

User Action:
Correct the error and resubmit the input file.

0067 DUPLICATE NAME

Explanation:
Names must be unique throughout the dictionary file. A name cannot be used to describe more than one entity.

User Action:
Select a name that is not already defined in the dictionary file. You can use the DDXREF command to produce a listing of all names in the dictionary file.

0068 LINE DEFINITION MUST CONTAIN AT LEAST ONE FIELD

Explanation:
Each line contains one or more fields that describe the characteristics of the data in a file. The field length must be from 1 to 255 characters.

User Action:
Add a field ID to the line definition and resubmit the input file.

0069 DUPLICATE TAG VALUE

Explanation:
Each line definition in the file must contain unique tag values. This enables the data manager to distinguish between the different line types.

User Action:
Correct the error and resubmit the input file.

0070 FIELD LENGTH EXCEEDS 255 CHARACTERS

Explanation:
The maximum length of a single field is 255 characters.

User Action:
Reduce the length of the field to 255 characters. If the field is actually longer than 255 characters, break the field into two or more fields.

0071 GROUP LENGTH EXCEEDS 255 CHARACTERS

Explanation:
The maximum length of a single group is 255 characters.

User Action:
Delete some fields from the group definition until the size is less than 255 bytes.

0072 KEY LENGTH EXCEEDS 40 CHARACTERS

Explanation:
The maximum length of a key for a data base file is 40 characters.

User Action:
Reduce the size of the field or select another field to be the key.

0073 INPUT FILE IO ERROR - ?1

Explanation:
An operating system error occurred while the input file was being read.

User Action:
Consult the operating system error message manual for a complete explanation of the two-character error code.

0074 GROUP DEFINITION MUST CONTAIN AT LEAST ONE FIELD

Explanation:
A group is a collection of one or more fields. A group must contain at least one field.

User Action:
Add at least one field to the group definition.

8.3 IDL ERROR MESSAGES

The interactive data librarian generates the following messages.

0101 DUPLICATE NAME.

Explanation:
The name specified was not unique to the dictionary.

User Action:
Correct the name or enter a new unique name in the dictionary.

0102 DUPLICATE ID.

Explanation:
The ID specified was not unique to the dictionary.

User Action:
Correct the ID or enter a new unique ID in the dictionary.

0104 FIRST CHARACTER MUST BE ALPHABETIC.

Explanation:
The first character in a name or ID must be a letter.

User Action:
Modify the name/ID so that the first character is a letter.

0105 INVALID ID.

Explanation:
The ID specified has not been defined to the dictionary.

User Action:
Correct the ID if it was misspelled or enter the new ID in the dictionary.

0106 INVALID NAME.

Explanation:
The name specified has not been defined to the dictionary.

User Action:
Correct the name if it was misspelled or enter the new name in the dictionary.

0107 FIELD SPECIFIED MORE THAN ONCE IN THE SAME GROUP DEFINITION.

Explanation:
The field specified was not unique to the group definition.

User Action:
Correct the field if it was misspelled or enter a unique field to the group definition.

0108 UNDEFINED ENTITY.

Explanation:
The entity specified was not defined to the dictionary.

User Action:
Correct the entity name if it was misspelled or enter the new entity in the dictionary.

0109 ONLY NAME AND DESCRIPTION CAN BE MODIFIED

Explanation:
Only the name and description of a previously defined entity can be modified if that entity is referenced by another entity in the dictionary.

User Action:
Modify only the name or description of the entity or create a new entity with the desired attributes.

0110 AT LEAST ONE FIELD IS REQUIRED IN A GROUP DEFINITION.

Explanation:
At least one field is required within a group definition.

User Action:
Add a field to the group definition or abort the group definition.

0111 ENTITY MAY NOT BE DELETED

Explanation:
The entity specified is referenced by other dictionary entities and cannot be deleted.

User Action:
Delete those dictionary entities that refer to this entity or abort the entity deletion. The entity specified is referenced by other dictionary entities and cannot be deleted.

0112 REQUIRED FIELD.

Explanation:
You attempted to bypass a required field.

User Action:
Enter the required field.

0113 INVALID PASSWORD.

Explanation:
An invalid password was entered.

User Action:
Verify that a valid password is being used.

0116 PRIMARY KEY TYPE MUST BE "ID" FOR RELATIVE RECORD AND SEQUENTIAL FILES.

Explanation:
The primary key field of a relative record or sequential file must be a double precision integer.

User Action:
Enter a field ID or name of type ID as the primary key.

0117 PRIMARY KEY MUST BE INCLUDED IN LINE DEFINITION.

Explanation:
For KIFs, the primary key must be included in each line definition.

User Action:
Include the primary key, at the proper offset, in the line definition.

0118 PRIMARY KEY OFFSETS DO NOT MATCH.

Explanation:
For KIFs, the primary key in each line must be at the same offset.

User Action:
Correct the primary key so that it is at the proper offset in the line being defined.

0119 INVALID TAG SPECIFICATION.

Explanation:
An invalid tag field was specified.

User Action:
Choose a field which is an IS or CH data type. The tag field must begin at the same off-set in each line type defined for a file.

0120 INVALID LINE.

Explanation:
The tag field specified is not from the line type designated.

User Action:
Enter a tag field from the designated line type.

0124 AT LEAST ONE LINE DEFINITION MUST BE SPECIFIED.

Explanation:
No line was specified for a file definition.

User Action:
Specify a line during file definition.

0125 FIELD OR GROUP SPECIFIED MORE THAN ONCE IN THE SAME FILE DEFINITION.

Explanation:
The field or group specified was not unique to the file definition.

User Action:
Correct the field/group if it was misspelled or enter a new field/group in the dictionary.

0126 AT LEAST ONE FILE NAME OR ID MUST BE ENTERED.

Explanation:
At least one file name or ID must be specified when defining a program in the dictionary.

User Action:
When defining a program, specify at least one file name or ID that the program references.

0127 DATA DICTIONARY FILE KEY AREA IS FULL.

Explanation:
The key area of the dictionary is full.

User Action:
Execute a DDSTAT on the dictionary to find out which area of the dictionary is full. After saving the information in the current dictionary with the DDSAVE utility, create a new dictionary that can handle the increased number of entities. Then restore the information from the old dictionary using the DDRSTR utility.

0128 DATA DICTIONARY FILE DATA AREA IS FULL.

Explanation:
The data area of the dictionary is full.

User Action:
The user should execute a DDSTAT on the dictionary to find out which area of the dictionary has become full. After saving the information in the current dictionary with the DDSAVE utility, a new dictionary shouold created which is able to handle the increased number of entities. The information from the old dictionary should then be restored using the DDRSTR utility.

0129 DD ACCESS ERROR — ?1

Explanation:
The data manager encountered an error during an access to the dictionary.

User Action:
Check the list of two-character data manager error messages (paragraph 8.5) for a complete explanation of this particular error.

0131 KEY LENGTH EXCEEDS 100 CHARACTERS.

Explanation:
You attempted to enter a KIF key field with a length greater than 100 characters.

User Action:
Limit the length of all KIF keys to 100 characters or less.

0132 KEY LENGTH EXCEEDS 40 CHARACTERS.

Explanation:
You attempted to enter a data base key field with a length greater than 40 characters.

User Action:
Limit the length of all data base keys to 40 characters or less.

0133 MODIFIABLE AND NON-MODIFIABLE KEYS MAY NOT OVERLAP, SEE ?1, ?2.

Explanation:
An overlapping key has been defined for a KIF that has both modifiable and non modifiable parts.

User Action:
Make the keys consistent concerning their modifiability or make them non overlapping.

0134 A MAXIMUM OF 13 SECONDARY KEYS MAY BE DEFINED.

Explanation:
A maximum of 13 secondary keys are allowed for data base files and KIFs.

User Action:
Limit the number of secondary keys to 13.

0135 TEMP FILE IO ERROR — ?1

Explanation:
An SVC error has occurred during an access to a temporary file.

User Action:
Refer to the operating system error message manual for a complete explanation of the two-character error code.

0136 TAG DEFINITIONS ARE REQUIRED BEFORE THIS FILE CAN BE ENTERED.

Explanation:
If you entered Y in response to the prompt TAGS INCLUDED IN LINES? (Y/N) during file definition, you must enter tag fields.

User Action:
Enter the appropriate tag fields or reverse the tag option on the file definition screen.

0137 AT LEAST ONE SECONDARY KEY MUST BE DEFINED BEFORE THIS FILE IS ENTERED.

Explanation:
If you entered Y in response to the prompt ANY SECONDARY KEYS? (Y/N) at least one secondary field must be entered.

User Action:
Enter the appropriate secondary keys or reverse the secondary key option on the file definition screen.

0138 DEFINED FILE TYPE MAY NOT BE CHANGED.

Explanation:
You attempted to change the type of a file already defined to the dictionary.

User Action:
Delete the file definition and reenter it with the new file type.

0139 TAG TYPE MUST BE DATA TYPE "IS" OR "CH".

Explanation:
You attempted to select a tag field that is not of type character or integer.

User Action:
Select a tag field of type IS or CH.

0142 VALUE IS TOO LARGE FOR INTEGER DATA TYPE.

Explanation:
The value specified for a numeric tag field exceeds the limits of the integer data type.

User Action:
Limit the value of an integer tag field to less than 65535.

0143 NON-NUMERIC CHARACTERS ENTERED FOR NUMERIC DATA TYPE.

Explanation:
The value specified for a tag field of type integer contains non numeric characters.

User Action:
Enter a numeric value for the tag value.

0144 DUPLICATE TAG VALUE.

Explanation:
The value for a tag field must be unique in a file.

User Action:
Enter a unique value for that tag field.

0145 DATA DICTIONARY FILE IS NOT ASSIGNED.

Explanation:
The IDL cannot be executed without an assigned dictionary.

User Action:
Assign a dictionary and then restart the IDL.

0147 INCOMPLETE ENTITY — CONSULT THE MANUAL BEFORE PROCEEDING.

Explanation:
An unknown condition caused the IDL to abnormally terminate during a previous execution when this file or group was being defined. The definition in the dictionary for this entity is incomplete.

User Action:
Any file or group being defined should be deleted and reentered.

0148 AT LEAST ONE FIELD OR GROUP IS REQUIRED IN A LINE DEFINITION.

Explanation:
A line may not be defined without a field or group defined in it.

User Action:
Define a field or group to the line or abort the line definition.

0149 NAME MAY NOT TERMINATE WITH A DOLLAR SIGN, DASH, OR UNDERSCORE CHARACTERS.

Explanation:
The name may not end with a dollar sign ($), a dash (-), or an underscore (_).

User Action:
Correct the name and reenter.

0150 A000 MAY NOT BE DELETED FROM THE DATA DICTIONARY.

Explanation:
A000 is a reserved ID used as the default primary key field for relative record and sequential files. It may not be deleted.

User Action:
Abort field modification.

8.4 DD-990 UTILITIES ERROR MESSAGES

The following error messages are generated by the DD-990 utilities:

0200 DD ACCESS ERROR — ?1

Explanation:
The data manager encountered an error during an access to the dictionary.

User Action:
Check the list of two-character data manager error messages (paragraph 8.5) for a complete explanation of this particular error.

0201 DATA DICTIONARY FILE IS NOT ASSIGNED

Explanation:
No dictionary was assigned when the utility was executed.

User Action:
Assign a dictionary and retry the utility.

0202 INVALID PASSWORD

Explanation:
An invalid password was entered.

User Action:
Verify that a valid password is being used.

0203 INPUT FILE ACCESS ERROR — ?1

Explanation:
An SVC error has occurred during an access to the input file.

User Action:
Consult the operating system error message manual for a complete explanation of the two-character error code.

0204 INPUT FILE CANNOT BE DUMY

Explanation:
A valid file must be specified for input.

User Action:
Enter the pathname of the valid input file.

0205 LISTING ACCESS ERROR — ?1

Explanation:
An SVC error has occurred during an access to the listing file.

User Action:
Consult the operating system error message manual for a complete explanation of the two-character error code.

0206 UNSUPPORTED DATA TYPE — ?1

Explanation:
The file contains a field whose data type is not supported by COBOL.

User Action:
Edit the file definition so that it does not contain fields of the specified data type and retry.

0207 NAME/ID MUST BE A FILE ENTITY

Explanation:
The name/ID entered must be a valid file defined to the dictionary.

User Action:
Enter a valid file name/ID that has been defined to the dictionary.

0208 INTERNAL DICTIONARY STRUCTURES DESTROYED

Explanation:
The internal structures in the data dictionary have been destroyed, possibly due to hardware errors or a system crash.

User Action:
If possible, execute DDSAVE, recreate the dictionary file, and execute DDRSTR to recover the dictionary file. If this is unsuccessful, delete the dictionary file, recreate it, and repopulate it.

0209 ** UNABLE TO FIND ENTITY NAME/ID — ?1

Explanation:
The entity name/ID specified is not in the dictionary.

User Action:
Verify the entity name/ID and make sure the appropriate dictionary is assigned.

0210 NAME/ID MUST BE A FILE, GROUP OR FIELD ENTITY

Explanation:
The entity name/ID specified must be a file, group, or field.

User Action:
Verify the entity name/ID and make sure the appropriate dictionary is assigned.

0211 INVALID ENTITY TYPE — ?1

Explanation:
An invalid entity type has been specified. Valid entity types are FL, GR, FI, and PR.

User Action:
Enter a valid entity type.

0212 DATA DICTIONARY FILE IS NOT ASSIGNED

Explanation:
This utility may not be executed without a dictionary being assigned.

User Action:
Assign the appropriate dictionary.

0213 DATA DICTIONARY FILE DATA AREA IS FULL

Explanation:
The data area of the dictionary is full.

User Action:
Execute a DDSTAT on the dictionary to find out which area of the dictionary is full. After saving the information in the current dictionary with the DDSAVE utility, create a new dictionary that can handle the increased number of entities. Then restore the information from the old dictionary using the DDRSTR utility.

0214 DATA DICTIONARY FILE KEY AREA IS FULL

Explanation:
The key area of the dictionary is full.

User Action:
Execute a DDSTAT on the dictionary to find out which area of the dictionary is full. After saving the information in the current dictionary with the DDSAVE utility, create a new dictionary that can handle the increased number of entities. Then restore the information from the old dictionary by using the DDRSTR utility.

0215 CANNOT FIND ENTITY ID — ?1

Explanation:
The entity ID specified cannot be found in the dictionary.

User Action:
Verify the entity ID and make sure the appropriate dictionary is assigned.

0216 CANNOT FIND FIELD ID — ?1

Explanation:
The field ID specified cannot be found in the dictionary.

User Action:
Verify the field ID and make sure the appropriate dictionary is assigned.

0217 CANNOT FIND FILE ID — ?1

Explanation:
The file ID specified cannot be found in the dictionary.

User Action:
Verify the file ID and make sure the appropriate dictionary is assigned.

0218 LISTING DIRECTORY NAME REQUIRED

Explanation:
The list fill utility requires a directory pathname when no ID is specified.

User Action:
Enter a directory pathname for the listing access name.

8.5 DATA MANAGER ERROR MESSAGES

The data manager generates the following error messages.

Code	Type of Error	Probable Cause
AC	ACCESS	You attempted to access a file using an improper access type. This can result from trying to close a file that has not been opened or to open a file with an undefined access type.
AE	ADDRESS	You supplied an invalid address in loc1 or loc2. This error can occur if you accidentally alters loc1 or loc2. For KIFs, a currency error has occurred. For relative record and sequential file files an invalid record number has been specified

Code	Type of Error	Probable Cause
AS	ADD	You are attempting to execute an add command with loc1 not set to asterisks.
BF	BAD FILE	The dictionary has a bad internal pointer or address. This can result if a system failure occurred while the dictionary was being modified. Rebuild the dictionary using DDSAVE/DDRSTR.
DA	DELETE ASTERISKS	A delete record (DR) function has been specified and loc1 does not contain asterisks.
DB	DATA BASE	The data base is not running.
DD	DATA DICTIONARY	The data manager is not running.
DF	DUPLICATE FILE	An attempt has been made to assign a file ID that has already been assigned.
DI	DICTIONARY I/O	The operating system has encountered an I/O error while reading or writing to the data dictionary. Check the system log for the exact SVC error code.
DK	DICTIONARY KEY	To correctly access the dictionary, you must specify both random and sequential key access methods when generating DBMS-990.
DL	DATA AREA LENGTH	The data area specified in a call to DD-990 is too small to accommodate the requested information.
DM	DATA DICTIONARY	The data manager is not running.
DU	DUPLICATE	You attempted to add an invalid record. For a KIF, the request specified a key value already in the file and that key field is defined as being non-duplicatable. For a relative record file the request specified a record number already in the file.
FA	FIND ASTERISKS	The asterisks at the end of the line list were not found. This can result if the asterisks do not start on a word boundary or if they are not the correct length.
FD	FILE DESCRIPTION	You attempted to assign a file whose physical type differs from the type defined for that file in the dictionary.

Code	Type of Error	Probable Cause
FE	FIELD	You specified an invalid field or group ID(s) in the parameter list. DD-990 cannot find the field or group ID.
FI	FILE	The file ID specified is not in the dictionary.
FN	FUNCTION	An invalid function code was specified. The function passed to DD-990 in the control block is undefined.
FT	FILE TYPE	DD-990 is unable to determine the file type from the dictionary.
IA	INVALID ACCESS	You attempted to insert or update a sequential file that has not been opened with exclusive access.
IE	INVALID ENTRY ID	The key ID specified is not the primary or secondary key. For an add or delete function, the key ID must be the primary key.
IG	INVALID GROUP	The group ID specified for a query group (QG) function is not a group.
II	INVALID ITEM	The user is not authorized to perform the function against a data item specified in the call.
IK	INVALID KEY	Either the data line to which loc1 points does not contain the same value as the key value given in the control block, or you attempted to execute a read on a secondary key that does not exist in the line specified.
IL	INVALID LINE	The specified line type does not contain the specified field.
IO	I/O	The operating system has encountered an I/O error on reading or writing to the disk. Check the system log for the exact SVC error code.
KC	KIF CURRENCY	DD-990 has detected a KIF currency error. Check the system log for the exact SVC error code.
KD	KEY DEFINITION	You attempted to assign a KIF file whose physical keys do not match the keys defined for that file in the dictionary.
KF	KIF FILE	You attempted to define a KIF when KIF access was not generated into DD-990.

Code	Type of Error	Probable Cause
KU	KEY UPDATE	You attempted to alter a KIF field that is defined in the dictionary as nonmodifiable.
KV	KEY VALUE	You attempted to access a relative record or sequential file using a key value that is not a valid record number.
LA	LINE ASTERISKS	A multiple line specification has been passed to DD-990 but no asterisk was found for a write (WT) or delete (DL) function.
LE	LINE	DD-990 has received an invalid line list. The LINE= syntax cannot be located. Thus, the field or group IDs cannot be found.
LK	LINE LOCKED	DD-990 has attempted to access a KIF record that has been locked by another user.
LN	LENGTH	You attempted to specify a field length larger than 256 bytes.
NB	NO BUFFER	Not enough buffers are available to facilitate the required operation.
ND	NO DICTIONARY	No dictionary has been assigned to DD-990.
NF	NO FILE	DD-990 cannot find the file ID in the file command. The file ID may be misspelled or it may have been released.
NH	NO HOLD	You attempted to delete or update a line that has not been held. Prior to deleting or updating a line, a read with hold option must have been specified.
NK	NO KEY	DD-990 cannot find the primary or secondary key value for a read forward (RF) or a read backward (RB) function.
NM	NO MEMORY	DD-990 cannot get enough memory to handle the request. Release one of the assigned files to obtain more memory.
OA	OPEN ASSIGN LUNO	An operating system error occurred while DD-990 was trying to assign a LUNO to the file. Check the system log for the exact SVC error code.

Code	Type of Error	Probable Cause
OE	OPEN ERROR	An operating system error occurred while you were trying to open the file. The file may already be in use or it may not exist. Check the system log for the exact SVC error code.
PL	PATHNAME LENGTH	You attempted to assign a file whose physical location is specified by an invalid pathname. Verify that the pathname is less than 48 characters long.
RL	RECORD LENGTH	If a data base file was assigned, the record length is not a valid page size. For a conventional file, the offset of the field specified is greater than the logical record length of the file.
RR	RELREC	You attempted to define a relative record file when that file access was not generated into DD-990.
S1	SHRD OPEN	You are attempting to open a file with EXCL access while the file is already open with SHRD access, or the same task has already opened the file with SHRD access.
SA	SHRD ACCESS	You attempted to open a file with EXCL access when it was assigned with SHRD access.
SC	SECURITY VIOLATION(DD)	You entered an invalid password and are not allowed to access the dictionary.
SQ	SEQUENTIAL	You attempted to define a sequential file to DD-990 when sequential file access was not generated into DD-990.
SV	SECURITY VIOLATION	You entered an invalid password and are not authorized to use the data item specified in the call.
TF	TAG FIELDS	You attempted to assign a file that contains tag fields that are undefined in the dictionary for that file.
UF	UNDEFINED FIELD	The field name specified in the line list is not contained in the dictionary definition of that file.
UL	UNDEFINED LINE	The line type specified for this file is not contained in the dictionary definition of that file.

Code	Type of Error	Probable Cause
WL	WRONG LINE TYPE	You attempted to insert a line that contains a tag field, and the value for that tag field is different from that listed in the definition of that line in the dictionary.
X1	EXCL OPEN	You are attempting to open a file with EXCL or SHRD access when the file is already open with EXCL access.

0250 INVALID FUNCTION

Explanation:
The DMFUNC task was bid with an invalid function code.

User Action:
Check to see that the SCI procedure has not been altered.

0251 INVALID PATHNAME

Explanation:
An invalid pathname was entered.

User Action:
Retry, using a valid pathname.

0252 INVALID STATUS CODE — ?1

Explanation:
The data base manager returned an error status code.

User Action:
Refer to the error messages listed in paragraph 8.5 for a complete explanation of this error.

0253 INVALID FILE

Explanation:
An invalid file name or ID was specified.

User Action:
Retry, using a file name or ID defined in the dictionary.

0254 ILLEGAL FILE ACCESS

Explanation:
An illegal access mode was selected on an assign file ID.

User Action:
Retry the assign file using either SHRD or EXCL access.

0261 UNABLE TO GET PARAMETERS

Explanation:
A parameter required by the data manager is either missing or invalid.

User Action:
Verify that the parameters used in the bid are correct.

0262 UNABLE TO OPEN DICTIONARY FILE

Explanation:
The data manager is unable to open the dictionary file. Some other task may be executing with exclusive access to the dictionary.

User Action:
Retry the operation when the dictionary file has been released from exclusive access.

0263 UNABLE TO GET MEMORY

Explanation:
The data manager cannot get any more memory.

User Action:
Release one of the assigned files to free sufficient memory for the current request.

8.6 SDM ERROR MESSAGES

The Start Data Manager (SDM) command generates the following error messages:

0271 DATA MANAGER ALREADY RUNNING

Explanation:
The data manager is already running.

User Action:
No user action is required.

0272 INVALID OR MISSING PARAMETER

Explanation:
An error occurred when trying to obtain parameters.

User Action:
The task was bid with the wrong parameters. Check to see that the SCI procedure has not been altered.

0273 UNABLE TO BID DATA MANAGER. SVC ERROR — 2B?1

Explanation:
An SVC error occurred during the bid task.

User Action:
Refer to the SVC error code for the correct action.

8.7 DDSTAT ERROR MESSAGE

The following message is generated by the DDSTAT procedure:

0301 DDSTAT UNABLE TO GET PARMS

Explanation:
An error occurred when trying to obtain parameters.

User Action:
The task was bid with the wrong parameters. Check to see that the SCI procedure has not been altered.

8.8 AFD ERROR MESSAGE

The messages generated by the Automatic File Definition (AFD) utility can be either warnings or fatal errors. The AFD error warning messages are as follows:

UNRECOGNIZED LINE.

Explanation:
AFD is unable to find a line construct in the FD entry.

User Action:
Define the file using IDL or BDL.

UNRECOGNIZED-NAME CLAUSE IGNORED.

Explanation:
AFD ignores 88 level fields since DD-990 DDL has no corresponding construct.

User Action:
No user action is required for this warning error message.

REDEFINES FOR "<field/group name>" IGNORED.

Explanation:
DD-990 DDL does not have a redefines capability. Therefore, any data names in a REDEFINES statement are translated as comments.

User Action:
The data names that have been translated to commented DDL statements can be included in the dictionary file in one of two ways. A new line type can be added to the DDL that includes those names and also includes new names for the remainder of the line. Alternatively, the data names that are being redefined can be removed from the DDL and the redefined names can be used instead.

ERROR IN FIELD SIZE "<picture>".

Explanation:
A COMP-1 field has been found with a size greater than 5. A single-word integer of type IS/2 has been generated.

User Action:
No user action is required.

The AFD fatal error messages are as follows:

LINE "<line number>" DOES NOT CONTAIN THE PRIMARY KEY.

Explanation:
DD-990 DDL requires that the primary key be defined at the same locaton with the same name in each line type in a KIF. AFD has found a COBOL data definition that does not meet this requirement and translates the data names under that 01 level as a comment.

User Action:
If the data names that have been translated as comments are needed in the dictionary file, the line definition should be modified to include the primary name.

QUALIFICATION FOR "<field/group name>" IGNORED.

Explanation:
An IN or OF qualifier has been found and ignored.

User Action:
Be sure that the correct fields and/or groups are specified in the ID and SECONDARY_REFERENCES statements.

RENAMES CLAUSE IGNORED.

Explanation:
Either a 66 level entry has been found for a group or one has been found that renames a field not immediately preceding the 66 level.

User Action:
If any names that should be included in the dictionary are treated as comments, add the appropriate NAME clause to the DDL.

GROUP OCCURS FOR "<group name>" IGNORED.

Explanation:
A group entry containing an OCCURS clause has been found. DD-990 DDL has no construct for repeated groups.

User Action
Add the fields and/or groups to the DDL that are required to make the DDL actually reflect the structure of the data file.

ERROR IN PIC CLAUSE "<clause>".

Explanation:
AFD has found an invalid PICTURE clause. Either a numeric field contains some character other than 9, S, or V or an alphanumeric field contains some character other than X or A.

User Action:
Add the FIELD statement to the DDL with the appropriate DD-990 data type.

ERROR IN TYPING "<picture>".

Explanation:
A COMP type has been found that cannot be translated to DD-990 DDL.

User Action:
Add the FIELD statement to the DDL with an appropriate DD-990 data type.

[PRIMARY/ALTERNATE] KEY IS A COMMENTED GROUP.

Explanation:
Since DD-990 does not allow nested groups, a group has been translated as a comment that is also a key. Therefore, a group name that is specified in the ID or SECONDARY-REFERENCES statement does not appear in any of the file's line definitions.

User Action:
Remove some higher-level group(s) so that the key group can be included in a line definition.

FILE "<file name>" NOT FOUND.

Explanation:
No SELECT statement or FD entry can be located for the file specified in response to the FILE NAME prompt.

User Action:
Execute AFD again, specifying a file name that has a corresponding SELECT statement and FD entry in the COBOL source.

Appendix A

DD-990 Syntax and Key Words

A.1 INTRODUCTION

The DD-990 data definition language (DDL) contains key words and user definitions. The user definitions can be written using DDL statements. The key words are discussed later in this appendix.

The seven DDL statements are as follows:

- FILE
- ID
- LINE
- FIELD
- GROUP
- SECONDARY-REFERENCES
- END

A.1.1 Optional DDL Features

There are three optional DDL features that allow for internal documentation. They are as follows:

- NAME
- DESCRIPTION
- Comments

Only a few syntax differences occur between conventional files and data base files. TAG, DUP, MOD, ENDK, and VALUE are not allowed for data base files. Also, DD-990 allows NAME and DESCRIPTION for data base files, but DBMS-990 does not.

A.1.2 FILE Statement

The first line of the DDL is the FILE statement. The FILE statement format for KIFs, sequential files, and relative record files is as follows:

FILE statement syntax:

FILE = <id>, TYPE = <KIF|SEQ|RR>,[TAG]

Example of a FILE statement:

FILE = DF14, TYPE = KIF, TAG

The following describes the components of the FILE statement:

FILE
: The file ID is a unique name from 1 to 4 characters long that identifies the file to the data dictionary file. The first character of the unique name must be alphabetic. The remaining characters can be either alphabetic or numeric.

TYPE
: The file type is KIF for key indexed files, SEQ for sequential files, and RR for relative record files.

TAG
: Tag indicates that tag field values are defined for each line in the file. If you do not specify TAG, the file cannot contain tags.

A.1.3 ID Statement

Descriptions of the ID statement format are as follows:

ID statement syntax:

ID = <id> = <data format>,[DUP] = {y|n}],[MOD = {y|n}]

Example of an ID statement:

ID = PRKY = IS, DUP = Y, MOD = Y

The following describes the components of the ID statement:

ID
: The ID is from 1 to 4 characters long and identifies the primary key. The ID is used to access the file through the primary key.

<data format>

The data format describes the data type of the primary key. Three types of formats are available. Type one format specifies the format code. Type two format specifies the format code and the field length. Type three format specifies the format code, field length, and number of decimal places. The data types of primary keys must be ID for relative record and sequential files. See Appendix B for detailed listings of the data types supported by DD-990. Examples of the three format codes are as follows:

```
TYPE ONE:    IS
TYPE TWO:    CH/5
TYPE THREE:  CN/5.1
```

DUP

DUP signifies whether duplicate values are allowed for the primary key. DUP applies only to KIFs.

MOD

MOD specifies whether the primary key is modifiable and applies only to KIFs.

A.1.4 LINE Statement

The LINE statement format for conventional files is as follows:

LINE statement syntax:

```
LINE = <line type>
.
. <field and group statements>
.
ENDL
```

Example of a LINE statement:

```
LINE=08
.
.
.
ENDL
```

The following describes the components of the LINE statement:

LINE

The line type is a two-character ID and can be alphabetic or numeric. The line type must be unique within the file definition.

ENDL

The END LINE statement indicates the end of a line specification.

A.1.5 FIELD Statement

FIELD statements are allowed only between LINE and ENDL, or GROUP and ENDG. The FIELD statement format for conventional files is as follows:

FIELD statement syntax:

FIELD = <field id> = <data format>,VALUE = <"tag field value">

Example of a FIELD statement:

FIELD = AB02 = IS,VALUE = "03'6"

The following describes the components of the FIELD statement:

FIELD
: The field ID identifies the field. The field ID is a unique name from 1 to 4 characters long that identifies the field to the dictionary file. The <data format> describes the data type of the field.

VALUE
: The value associated with the field defines the tag value. You must use single or double quotes to contain the tag value. To insert space in this value, enclose the blanks in whichever kind of quote (single or double) has not been used to contain the tag value. Quotes are optional for integers.

A.1.6 GROUP Statement

The GROUP and ENDG statements bracket a list of fields (or a single field). You cannot define groups within other groups. The GROUP statement format for conventional files is as follows:

GROUP statement syntax:

```
GROUP = <group id>
.
.
.
ENDG
```

Example of a GROUP statement:

```
GROUP = AB05
FIELD = AB02 = IS
FIELD = AB03 = CH/2
ENDG
```

The following describes the components of the GROUP statement:

GROUP
: The group ID identifies the group. The group ID is a unique name from 1 to 4 characters long that identifies the group to the data dictionary file. You can declare the group as a secondary key. Each GROUP statement requires at least one FIELD statement.

ENDG

The ENDG statement defines the end of a group. Any succeeding FIELD statements are not part of the preceding group definition. Succeeding group definitions are allowed.

A.1.7 SECONDARY-REFERENCES Statement

Only one SECONDARY-REFERENCES statement is allowed in the DDL for a file. The SECONDARY-REFERENCES statement allows you to define secondary keys in KIFs. Secondary keys are allowed for KIFs only. Only thirteen secondary keys are allowed. The syntax for defining secondary keys is as follows:

SECONDARY-REFERENCES statement syntax:

SECONDARY-REFERENCES
{<group id>|<field id>},[DUP = {y|n}],[MOD = {y|n}]

Example of a SECONDARY-REFERENCES statement:

ADDR, DUP = N, MOD = Y

The following describes the components of the SECONDARY-REFERENCES statement:

GROUP ID

The group ID is from 1 to 4 characters long and identifies the group to the dictionary file.

FIELD ID

The field ID is from 1 to 4 characters long and identifies the field to the dictionary file.

DUP

Dup specifies whether duplicate values are allowed for the key.

MOD

MOD specifies whether the secondary key is modifiable.

A.1.8 END. Statement

The END. statement is the last line of the DD-990 DDL. The END. statement format is as follows:

END. statement syntax:

END.

The following describes the END. statement component:

END.

The END. statement defines the end of the file definition.

A.1.9 NAME Syntax

The NAME feature can follow a FILE, ID, LINE, GROUP, or FIELD statement. This feature is not allowed in the SECONDARY KEY statement. NAME provides an alias capability. It allows entities to be referenced by something more meaningful than a four-character ID. The syntax for NAME is as follows:

```
NAME = <name>
```

The following describes the component of the NAME feature:

NAME
A NAME can be up to 30 characters long and must begin with a letter. The rest of the NAME can consist of any combination of letters, digits, dashes, underscores, or dollar signs.

A.1.10 DESCRIPTION Syntax

DESCRIPTION may be defined for files, lines, groups, or fields. You cannot use the DESCRIPTION feature in SECONDARY KEY statements. The syntax for DESCRIPTION is as follows:

```
DESCRIPTION
<descriptive text>
ENDD
```

The following describes the components of the DESCRIPTION feature:

DESCRIPTION
DESCRIPTION marks the beginning of descriptive text.

<descriptive text>
The descriptive text can be up to 160 characters long.

ENDD
The ENDD statement indicates the end of the description specification.

A.1.11 Comments

Any information following an asterisk (*) is considered a comment. Some examples of comments are as follows:

```
*This is a comment before a statement
GROUP = <group id>
*This is a comment after a statement
ENDG *This is a comment on the same line
```

A.2 KEY WORDS

The following list of key words applies to data base, key indexed, sequential, and relative record files:

ACCESS
DESCRIPTION
DUP
END.
ENDD
ENDG
ENDK
ENDL
FIELD
FILE
GROUP
ID
LINE
LINES
MOD
NAME
RANDOM
SECONDARY-REFERENCES
SEQUENTIAL
TAG
TYPE
VALUE
VOL

ACCESS
This key word applies to data base files and represents the data retrieval routine for the primary key. The routines may be either sequential access (SEQUENTIAL/1) or random access (RANDOM/1).

DESCRIPTION
This key word indicates in the syntax that the information to follow will be descriptive text.

DUP
This key word specifies a duplicate primary key and applies only to KIFs.

END.
This key word designates the end of the file definition.

ENDD
This key word designates the end of a statement description.

ENDG
This key word defines the end of a group. Any succeeding FIELD statements are not part of the last group definition. Succeeding group definitions are allowed.

ENDK
This key word defines the end of a group key.

ENDL
This key word indicates the end of a line specification.

FIELD
This key word is identified by the field ID. The field ID is a unique name from 1 to 4 characters long that identifies the field to the data dictionary.

FILE
This key word is identified by the file ID. The file ID is a unique name from 1 to 4 characters long that identifies the file to the data dictionary. It is used to access the file.

GROUP
This key word must have a unique ID. The group can be declared as a secondary key.

ID
This key word is a name from 1 to 4 characters long identifying the primary key. This ID is used to access the file through the primary key.

LINE
This key word must be unique within the file definition.

LINES
This key word represents the maximum number of lines defined for the file.

MOD
This key word specifies whether the secondary key is modifiable and applies only to KIFs.

NAME
This key word is a name from 1 to 30 characters long and is used to define the key word in the statement. The name is added when the file definition is added.

RANDOM
This key word represents the access retrieval routine for the primary key.

SECONDARY-REFERENCES
This key word represents the beginning of the secondary key description.

SEQUENTIAL
This key word represents the access retrieval routine for the primary key.

TAG
This key word indicates that tag field values are defined for each line in the file. If it is not specified, the file does not contain tags.

TYPE
This key word is KIF for key indexed files, SQ for sequential files, and RR for relative record files.

VALUE
This key word associates the field with the tag to define the tag field value.

VOL
This key word represents the maximum number of primary keys that can exist in the file at any time. Since the primary key is unique for each line, this maximum number also represents the number of records in the file.

Appendix B

DD-990 Function Keys and Data Types

In order to use the IDL procedure, the function keys and data types must be defined. The following is a list of the function keys and their appropriate use:

Key	Operation
F1	Roll up
F2	Roll down
F3	Inactive
F4	Inactive
F5	Scroll up
F6	Scroll down
F7	Delete
F8	Insert
CMD	Abort or return to main screen
ENTER	Enter data in dictionary file
PRINT	Print current screen
BACKFIELD	View data currently in dictionary file

F1 (Roll Up) Key
This key performs a roll up function. It is activated in the file, line, secondary keys, and tag screens.

F2 (Roll Down) Key
This key performs a roll down function. It is activated in the line, secondary keys, and tag screens.

F5 (Scroll Up) Key
This key performs a scroll up function. It is activated in the group and line screens.

F6 (Scroll Down) Key
This key performs a scroll down function. It is activated in the group and line screens.

F7 (Delete) Key
This key performs a delete function and is activated in all screens. A Y or N response is required to be sure of the delete function being selected. It is activated by all screens.

F8 (Insert) Key
This key performs an insert function. It is activated in the group and line screens.

CMD (Abort) Key
This key performs an abort function, giving a Y or N response to be sure of the abort function being selected. It is activated by all screens.

ENTER Key

This key updates the dictionary file with the information contained on the VDT. If required fields have not been answered, the system does not allow the ENTER option. It is activated by all screens.

PRINT Key

This key prints the information contained on the VDT. It is activated by all screens.

BACKFIELD Key

This key can be used to view all the information entered on some of the IDL screens.

The data types are as follows:

Code	Description	Example Formats
AN	Arithmetic without sign. Decimal places are allowed. Use zero for no decimal places. Use type 3 format.	AN/8.2 COBOL: PIC 9(6)V9(2) COMP. FORTRAN: <none> Pascal: <none>
AS	Arithmetic signed. Length (n) must include sign, and decimal places are allowed. Use zero for no decimal places. Use type 3 format.	AS/8.2 COBOL: PIC S9(5)V9(2) COMP. FORTRAN: <none> Pascal: <none>
CH	Character string. Length includes total characters. Decimal places are not allowed. Use type 2 format.	CH/20 COBOL: PIC X(20). FORTRAN: <A format> Pascal: PACKED ARRAY [1. .20] OF CHAR
CN	Character numeric. Decimal places are allowed. Use zero for no decimal places. Use type 3 format.	CN/6.2 COBOL: PIC 9(4)V9(2). FORTRAN: <none> Pascal: <none>
CS	Character numeric signed. Length (n) must include the sign. Decimal places are allowed. Use zero for no decimal places. Use type 3 format.	CS/8.5 COBOL: PIC S9(2)V9(5) FORTRAN: <none> Pascal: <none>
IS	Single-precision integer. Contained in one 16-bit word. Length (n) default is 2; if specified, it must be 2. Field may contain a sign. Use type 1 or 2 format.	IS/2 COBOL: PIC 9(5) COMP-1. FORTRAN: INTEGER*2 Pascal: INTEGER

Code	Description	Example Formats
ID	Double-precision integer. Contained in two 16-bit words and may be signed. Length (n) default is 4; if specified, it must be 4. Use type 1 or 2 format.	ID/4 COBOL: <none> FORTRAN: INTEGER*4 Pascal: LONGINT
LG	Logical variable. Length (n) default is 2; if specified, it must be 2. Use type 1 or 2 format.	LG/2 COBOL: <none> FORTRAN: LOGICAL Pascal: BOOLEAN
PK	Packed decimal. Digit length (n) must be even and includes the sign. Decimal places are allowed, and zero indicates no decimal places. Contained in n/2 bytes. Use type 3 format.	PK/6.2 COBOL: PIC S9(3)V9(2) COMP-3. FORTRAN: <none> Pascal: <none>
RS	Single-precision real. Contained in two 16-bit words and may be signed. Length (n) default is 4; if specified, it must be 4. Use type 1 or 2 format.	RS/4 COBOL: <none> FORTRAN: REAL *4 Pascal: REAL
RD	Double-precision real. Contained in four 16-bit words and may be signed. Length (n) default is 8; if specified, it must be 8. Use type 1 or 2 format.	RD/8 COBOL: <none> FORTRAN: REAL *8 Pascal: <none>

Appendix C

DD-990 Test File Listings

The three conventional files listed in this appendix are used in examples throughout the manual. Figure C-1 is the listing for a relative record file named INVT. Figure C-2 is the listing for a key indexed file (KIF) named INVO. Figure C-3 is the listing for a sequential file named CUST.

INVENTORY-FILE (INVT)

LINE FORMAT:

PRIMARY KEY = PTNO (PART NUMBER)

DESC	PRIC	QUOH
15	6	5

FIELD LENGTHS

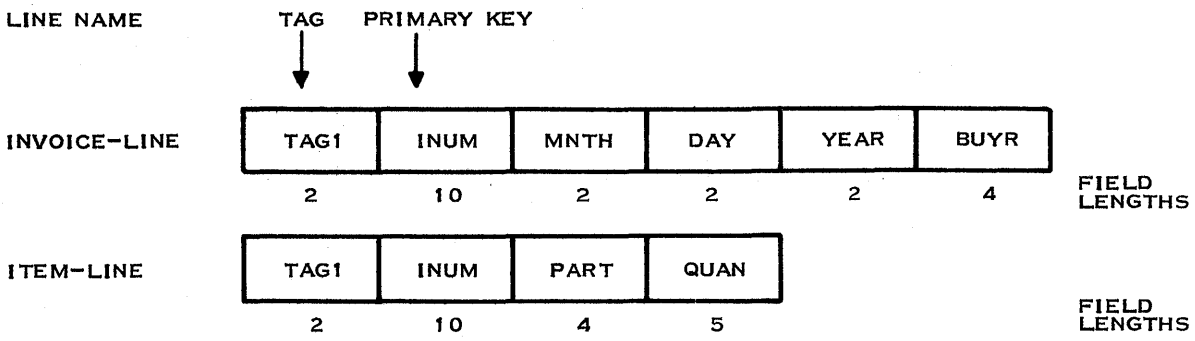
RECORD #	FILE CONTENTS		
0001	HAMMER	5.00	2000
0002	PLIERS	10.00	1500
0003	SCREW DRIVER	3.00	12000
0004	LEVEL	50.00	100
0005	WRENCH	20.00	500
0006	MICROMETER	50.00	200
0007	SAW	12.00	400

2280561

Figure C-1. Relative Record File

INVOICE-FILE (INVO)

LINE FORMATS:



FILE CONTENTS:

01	0000138820	03	15	78	0001
02	0000138820	0001	00200		
02	0000138820	0002	00500		
02	0000138820	0003	00700		
01	0000154588	04	28	78	0002
02	0000154588	0004	00020		
02	0000154588	0001	00100		
02	0000154588	0005	00050		
02	0000154588	0002	00050		
01	0000163023	05	25	78	0003
02	0000163023	0005	00100		
02	0000163023	0002	00300		
02	0000163023	0006	00040		
01	0000175571	05	28	78	0004
02	0000175571	0001	00400		
02	0000175571	0007	00100		

2280562

Figure C-2. Key Indexed File

CUSTOMER-FILE (CUST)

LINE FORMAT:

PRIMARY KEY = CSTN (CUSTOMER NUMBER)

NAME	STRE	CITY	STAT	ZIP

REC	FILE CONTENTS				
1	ABC HARDWARE STORE	8004 N. STREET	AUSTIN	TX	78750
2	HANDY ANDY HARDWARE	4000 BURNET ROAD	AUSTIN	TX	78753
3	CAPITOL HARDWARE CO.	8153 RESEARCH	AUSTIN	TX	78752
4	NORTHWEST HARDWARE	2000 LAMAR BLVD	AUSTIN	TX	78751

2280563

Figure C-3. Sequential File

Alphabetical Index

Introduction

HOW TO USE INDEX

The index, table of contents, list of illustrations, and list of tables are used in conjunction to obtain the location of the desired subject. Once the subject or topic has been located in the index, use the appropriate paragraph number, figure number, or table number to obtain the corresponding page number from the table of contents, list of illustrations, or list of tables.

INDEX ENTRIES

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections — Reference to Sections of the manual appear as "Sections x" with the symbol x representing any numeric quantity.

- Appendixes — Reference to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.

- Paragraphs — Reference to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph may be found.

- Tables — References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number.

  Tx-yy

- Figures — References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number.

  Fx-yy

- Other entries in the Index — References to other entries in the index preceded by the word "See" followed by the referenced entry.

CUT ALONG LINE

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer Data Dictionary User's Guide (2276582-9701)

Manual Date: 15 August 1982 **Date of This Letter:** _____

User's Name: _____ **Telephone:** _____

Company: _____ **Office/Department:** _____

Street Address: _____

City/State/Zip Code: _____

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

Location in Manual	Comment/Suggestion
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), TAPE AND MAIL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DIGITAL SYSTEMS GROUP
ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769

FOLD

TEXAS INSTRUMENTS
INCORPORATED
DIGITAL SYSTEMS GROUP
POST OFFICE BOX 2909 AUSTIN, TEXAS